21世纪高等学校规划教材 | 计算机应用



Web前端开发及应用教程

王丽铭 主编
唐哲卿 刘志凯 编著

清华大学出版社

21 世纪高等学校规划教材・计算机应用

Web 前端开发及应用教程

王丽铭　主　编
唐哲卿　刘志凯　编　著

清华大学出版社
北　京

内 容 简 介

Web 前端开发工程师是最近几年计算机行业的一个新兴岗位，而且未来发展的势头强劲，本书针对这一岗位的具体技术需求编写，内容涵盖了该岗位所需的基础及专业知识。全书共分五大部分 15 章，从平面设计到动画制作，从网页美工到网页动态技术，从易到难地将 Web 前端技术呈现在广大读者面前，每章以基础知识为主，综合练习为辅，从理论与实践两个方面对学生进行训练，以期学生最好地掌握课程内容，最快地适应工作岗位。

本书可以作为本、专科学校相关专业的教学用书，也可以作为想要从事 Web 前端开发职业的技术人员的学习用书。

图书在版编目（CIP）数据

Web 前端开发及应用教程/王丽铭主编. —北京：清华大学出版社，2016
21 世纪高等学校规划教材 • 计算机应用
ISBN 978-7-302-42052-1

Ⅰ. ①W… Ⅱ. ①王… Ⅲ. ①超文本标记语言－程序设计－高等学校－教材 ②网页制作工具－程序设计－高等学校－教材 Ⅳ. ①TP312 ②TP393.092

中国版本图书馆 CIP 数据核字（2015）第 263584 号

责任编辑：郑寅堃　王冰飞
封面设计：傅瑞学
责任校对：白　蕾
责任印制：

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn，010-62795954
印 刷 者：
装 订 者：
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　张**：22　　**字　　数**：534 千字
版　　次：2016 年 1 月第 1 版　　**印　　次**：2016 年 1 月第1 次印刷
印　　数：1～　000
定　　价：　.00 元

产品编号：065371-01

出版说明

随着我国改革开放的进一步深化，高等教育也得到了快速发展，各地高校紧密结合地方经济建设发展需要，科学运用市场调节机制，加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度，通过教育改革合理调整和配置了教育资源，优化了传统学科专业，积极为地方经济建设输送人才，为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是，高等教育质量还需要进一步提高以适应经济社会发展的需要，不少高校的专业设置和结构不尽合理，教师队伍整体素质亟待提高，人才培养模式、教学内容和方法需要进一步转变，学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007 年 1 月，教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》，计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”，通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容，进一步深化高等学校教学改革，提高人才培养的能力和水平，更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中，各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势，对其特色专业及特色课程(群)加以规划、整理和总结，更新教学内容、改革课程体系，建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上，经教育部相关教学指导委员会专家的指导和建议，清华大学出版社在多个领域精选各高校的特色课程，分别规划出版系列教材，以配合“质量工程”的实施，满足各高校教学质量和教学改革的需要。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作，提高教学质量的若干意见》精神，紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”，在有关专家、教授的倡议和有关部门的大力支持下，我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”)，旨在配合教育部制定精品课程教材的出版规划，讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师，其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求，“编委会”一致认为，精品课程的建设工作从开始就要坚持高标准、严要求，处于一个比较高的起点上；精品课程教材应该能够反映各高校教学改革与课程建设的需要，要有特色风格、有创新性(新体系、新内容、新手段、新思路，教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展，顺应并符合 21 世纪教学发展的规律，代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐)，经“编委会”认真评审，最后由清华大学出版

社审定出版。

目前，针对计算机类和电子信息类相关专业成立了两个“编委会”，即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色精品教材包括：

（1）21世纪高等学校规划教材·计算机应用——高等学校各类专业，特别是非计算机专业的计算机应用类教材。

（2）21世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。

（3）21世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。

（4）21世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。

（5）21世纪高等学校规划教材·信息管理与信息系统。

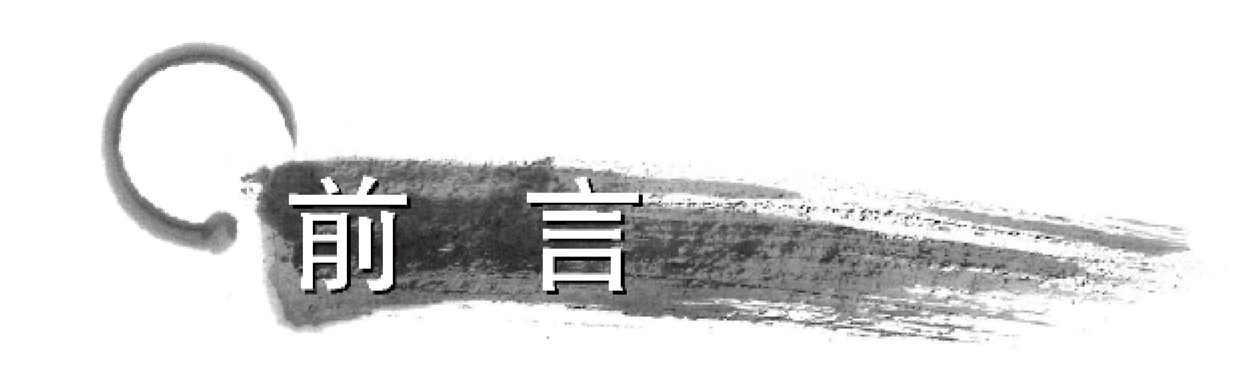
（6）21世纪高等学校规划教材·财经管理与应用。

（7）21世纪高等学校规划教材·电子商务。

（8）21世纪高等学校规划教材·物联网。

清华大学出版社经过三十多年的努力，在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌，为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格，这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会
联系人：魏江江
E-mail：weijj@tup.tsinghua.edu.cn



前 言

随着互联网从 Web 1.0 发展到 Web 2.0，网站也在静态页面(用户使用网站的行为也以浏览为主)的基础上添加了各种桌面软件，使网页不再只是承载单一的文字和图片，各种富媒体让网页的内容更加生动，网页上软件化的交互形式为用户提供了更好的使用体验，这些都是基于前端技术实现的，其中包括 CSS、HTML、DOM、Ajax、JavaScript 等。

Web 前端开发是一个很特殊的领域，涵盖的知识面非常广，既有具体的技术，也有抽象的理念。简单地说，它的主要职能就是把网站的界面更好地呈现给用户。Web 前端开发工程师除了要具有良好的团队合作精神以外，还要有不断学习的目标，在知识层面上包括从最初的 Photoshop、Flash、Dreamweaver 到现在常用的 HTML、CSS、JavaScript、Ajax 等技术。本书最真实地贴近 Web 前端工程师的岗位需求，从基础知识抓起，理论与实践相结合，在注重基础理论讲解的同时加强对动手能力的训练。

本书共分五大部分，第一部分只包括 1 章，是对 Web 知识的整体叙述；第二部分是 Photoshop 部分，对 Photoshop 进行了详细讲解；第三部分是 Flash 部分，对 Flash 的常用知识进行了讲解，包括工具箱的使用、动画的制作；第四部分是 HTML＋CSS 部分，对 HTML 语言、CSS 用于网页美化及布局进行了详细的讲解；第五部分是 JavaScript，包括 JavaScript 和 Ajax、jQuery，对网页的前端技术进行了讲解。

本书由在校的相关专业教师编著，其中，第 1～5 章由唐哲卿编写，第 6、7 章由单莹莹编写，第 8、9、11、12 章由王丽铭编写，第 13、15 章由刘志凯编写，第 14 章由郭鑫编写，第 10 章由王金伟编写。这些教师都在学校从事最基础的 Web 开发课程的讲解，同时具有企业开发的经验，对 Web 前端有自己的认识和理解。希望这些知识对在校学生、想从事 Web 前端开发的读者有一定的指导和帮助作用。

由于编者的能力有限，本书还有很多不足之处，敬请广大读者给予指正。

编　者

2015 年 10 月

目 录

第1章 Web前端开发技术概述

本章学习目标：

✍了解 Web 的起源与发展。

✍了解 Web 前端工程师的岗位需求。

✍掌握 Web 前端开发需要的技术和相关工具。

1.1 Web 概述

Web 的本意是“蜘蛛网”和“网”，在网页设计中被称为“网页”，表现为 3 种形式，即超文本(Hypertext)、超媒体(Hypermedia)、超文本传输协议(HTTP)。

1.1.1 Web 的起源

最早的网络构想可以追溯到 1980 年蒂姆·伯纳斯-李构建的 ENQUIRE 项目，这是一个类似维基百科的超文本在线编辑数据库。尽管这与万维网大不相同，但是它们有许多相同的核心思想，甚至还包括一些伯纳斯-李的万维网之后的下一个项目“语义网”中的构想。

1989 年 3 月，伯纳斯-李撰写了《关于信息化管理的建议》一文，文中提及 ENQUIRE 并且描述了一个更加精巧的管理模型。1990 年 11 月 12 日，他和罗伯特·卡里奥(Robert Cailliau)合作提出了一个更加正式的关于万维网的建议。1990 年 11 月 13 日，他在一个 NeXT 工作站上写了第一个网页以实现他文中的想法。

在那年的圣诞假期，伯纳斯-李制作了一个网络工作必须的工具——第一个万维网浏览器(同时也是编辑器)和第一个网页服务器。

1991 年 8 月 6 日，他在 alt. hypertext 新闻组上贴了关于万维网项目简介的文章，这一天也标志着因特网上万维网公共服务的首次亮相。

万维网中至关重要的概念——超文本起源于 20 世纪 60 年代的几个项目。例如泰德·尼尔森(Ted Nelson)的仙那都项目(Project Xanadu)和道格拉斯·英格巴特(Douglas Engelbart)的 NLS，而这两个项目的灵感都来源于万尼瓦尔·布什在其 1945 年的论文《和我们想得一样》中为微缩胶片设计的“记忆延伸”(memex)系统。

蒂姆·伯纳斯-李的另一个才华横溢的突破是将超文本嫁接到因特网上。在他的《编织网络》一书中，他解释说他曾一再向这两种技术的使用者们建议它们的结合是可行的，但是却没有任何人响应他的建议，最后他只好自己解决了。他发明了一个全球网络资源唯一认

证的系统——统一资源标识符。

万维网和其他超文本系统有很多不同之处：

第一，万维网上需要单向连接而不是双向连接，这使得任何人可以在资源拥有者不做任何行动的情况下链接该资源。和早期的网络系统相比，这一点对于减少实现网络服务器和网络浏览器的困难至关重要，它的副作用是产生了坏链的慢性问题。

第二，万维网不像某些应用软件(如 HyperCard)，它不是私有的，这使得服务器和客户端能够独立地发展和扩展，且不受许可限制。

1993 年 4 月 30 日，欧洲核子研究组织宣布万维网对任何人免费开放，并不收取任何费用。两个月之后，Gopher 宣布不再免费，造成大量用户从 Gopher 转向万维网。

万维网联盟(World Wide Web Consortium，W3C)又称 W3C 理事会，1994 年 10 月在麻省理工学院计算机科学实验室成立，建立者是万维网的发明者蒂姆·伯纳斯-李。

1.1.2 Web 技术

Web 是一种典型的分布式应用架构。Web 应用中的每一次信息交换都要涉及客户端和服务端两个层面，因此 Web 开发技术大体上也可以被分为客户端技术和服务端技术两大类。

1. 客户端技术

1) HTML 语言的诞生

Web 客户端的主要任务是展现信息内容，HTML 语言是信息展现的最有效的载体之一。作为一种实用的超文本语言，HTML 的历史最早可以追溯到 20 世纪 40 年代。1969 年，IBM 公司的 Charles Goldfarb 发明了可用于描述超文本信息的 GML 语言。从 1978 到 1986 年，在 ANSI 等组织的努力下，GML 语言进一步发展成为著名的 SGML 语言标准。当蒂姆·伯纳斯-李在 1989 年试图创建一个基于超文本的分布式应用系统时意识到，SGML 过于复杂，不利于信息的传递和解析。于是，蒂姆·伯纳斯-李对 SGML 语言做了大刀阔斧的简化和完善。1990 年，第一个图形化的 Web 浏览器"World Wide Web"终于可以使用一种为 Web 度身定制的语言——HTML 来展现超文本信息了。

2) 从静态信息到动态信息

最初的 HTML 语言只能在浏览器中展现静态的文本或图像信息，随后由静态技术向动态技术逐步转变。Web 出现后，GIF 第一次为 HTML 页面引入了动感元素。1995 年，Java 语言的问世带来了更大的变革。Java 语言天生就具备平台无关的特点，让人们一下子找到了在浏览器中开发动态应用的捷径。CSS 和 DHTML 技术真正让 HTML 页面又酷又炫、动感无限起来。1997 年，Microsoft 公司发布了 IE 4.0，并将动态 HTML 标记、CSS 和动态对象模型发展成了一套完整、实用、高效的客户端开发技术体系，Microsoft 称其为 DHTML。同样是实现 HTML 页面的动态效果，DHTML 技术无须启动 Java 虚拟机或其他脚本环境，可以在浏览器的支持下获得更好的展现效果和更高的执行效率。

为了在 HTML 页面中实现音频、视频等更加复杂的多媒体应用，又引入了对 QuickTime 插件的支持，插件这种开发方式迅速风靡了浏览器的世界，还有 20 世纪 90 年代中期刚刚问世的 COM 和 ActiveX 也十分流行。Real Player 插件、Microsoft 自己的媒体播

放插件 Media Player 也被预装到了各种 Windows 版本之中，随后 Flash 插件走入了人们的视线。

2. 服务端技术

与客户端技术从静态向动态的演进过程类似，Web 服务端的开发技术也是由静态向动态逐渐发展、完善起来的。

最早的 Web 服务器简单地响应浏览器发来的 HTTP 请求，并将存储在服务器上的 HTML 文件返回给浏览器。

第一种真正使服务器能根据运行时的具体情况动态生成 HTML 页面的技术是大名鼎鼎的 CGI 技术，CGI 技术允许服务端的应用程序根据客户端的请求动态生成 HTML 页面，这使客户端和服务端的动态信息交换成为了可能。

早期的 CGI 程序大多是编译后的可执行程序，其编程语言可以是 C、C++、Pascal 等任何通用的程序设计语言。为了简化 CGI 程序的修改、编译和发布过程，人们开始探寻用脚本语言实现 CGI 应用的可行方式。

1994 年，人们发明了专用于 Web 服务端编程的 PHP 语言。PHP 语言将 HTML 代码和 PHP 指令合成为完整的服务端动态页面，可以用一种更加简便、快捷的方式实现动态 Web 功能。1996 年，Microsoft 公司在其 Web 服务器 IIS 3.0 中引入了 ASP 技术，ASP 使用的脚本语言是用户熟悉的 VBScript 和 JavaScript。1998 年，JSP 技术诞生。

随后，XML 语言及相关技术成为主流。XML 语言对信息的格式和表达方法做了最大程度的规范，应用软件可以按照统一的方式处理所有 XML 信息，这样一来，信息在整个 Web 世界里的共享和交换就有了技术上的保障。HTML 语言关心的是信息的表现形式，而 XML 语言关心的是信息本身的格式和数据内容。

1.1.3　Web 技术的发展

1. Web 技术发展的第一阶段——静态技术阶段

本阶段的 Web 主要是静态的 Web 页面。在这个阶段，HTML 语言就是 Web 向用户展示信息的最有效的载体。HTML 的全称是超文本标记语言（Hypertext Markup Language），它通过提供超文本格式的信息在客户端的用户机上显示出完整的页面。Web 服务器使用 HTTP（Hypertext Transport Protocol，超文本传输协议）将 HTML 文档从 Web 服务器传输到用户的 Web 浏览器上。

在本阶段，由于受 HTML 语言和旧式浏览器的制约，Web 页面只包含静态的文本和图像信息，限制了资源共享，这个现象越来越不能满足人们对信息多样性和及时性的要求。而这一阶段的 Web 服务器基本上只是一个 HTTP 的服务器，它负责接收客户端浏览器的访问请求，建立连接，响应用户的请求，查找所需的静态的 Web 页面，再返回到客户端。

2. Web 技术发展的第二阶段——动态技术阶段

在 Web 出现的同时，能存储、展现二维动画的 GIF 图像格式也发展成熟，为 HTML 引入动态元素提供了条件。此后，为了能更好地弥补静态页面的不足，人们将传统单机环境下

的编程技术引入互联网络与 Web 技术相结合，从而形成新的网络编程技术。1995 年 Java 语言的问世给 Web 的发展带来更大的变革，它为人们提供了一条在浏览器中开发应用的捷径。1996 年，著名的 Netscape 浏览器 2.0 版本和 Microsoft 的 IE 3.0 增加了对 JavaApplets 和 JavaScript 的支持。JavaScript 语言是一种以脚本方式运行的、简化的 Java 语言，Web 世界里从此出现了脚本技术。

其实，真正让 HTML 页面又酷又炫、动感无限的是 CSS(Cascading Style Sheets)和 DHTML(Dynamic HTML)技术。1996 年底，W3C 组织提出了 CSS 的建议标准，同年，IE 3.0 引入了对 CSS 的支持，这项技术使得开发者能够在 Web 上更好地把握信息的展示。1997 年的 Netscape 4.0 在支持 CSS 技术的同时又增加了许多由他们自定义的动态 HTML 标记，同年，Microsoft 公司发布了 IE 4.0，并将动态的 HTML 标记、CSS 和动态对象模型(DHTML Object Model)发展成了一套完整的客户端开发技术体系(DHTML)。该项技术无须启动 Java 虚拟机或其他脚本环境，在浏览器的支持下，同样可以实现 HTML 页面的动态展示，而且可以获得更好的效果。

1996 年，Netscape 2.0 成功引入了对 QuickTime 插件的支持，从此实现了在 HTML 页面下音频、视频等更加复杂的多媒体应用。同年，IE 3.0 正式支持在 HTML 页面中插入 ActiveX 控件的功能。从此，各种各样由不同公司所开发的插件先后在浏览器上取得了成功。

这里所说的动态页面和静态页面是相对应的，在引入了动态技术生成的网页中，网页 URL 的后缀不只是.htm、.html、.shtml、.xml 等静态网页的常见形式，还可以是.asp、.jsp、.php、.perl、.cgi 等。从网页内容的显示上看，动态网页引入了各项技术，使得网页内容更多样化，引人入胜；从网站的开发管理和维护角度看，动态网页以数据库技术为基础，更利于网站的维护，而动态网页使用了 ASP 对象，可以实现诸如用户注册、用户登录、数据管理等功能，提高了网络的利用率，为用户提供了更多的方便。

3. Web 技术发展的第三阶段——Web 2.0 新时期

在最近两年里，Web 2.0 这个名词引起了很多人的关注，什么是 Web 2.0 呢？其实，Web 2.0 并没有一个准确的定义，甚至它并不是一个具体的事物，而只是人们对于一个阶段的描述。在这一阶段，用户可以自己主导信息的生产和传播，从而打破了原先所固有的单向传输模式。Web 2.0 并不是一个革命性的改变，而只是应用层面的东西，相对于传统的门户网站，它具备了更好的交互性。Web 2.0 是以 Flickr、43Things.com 等网站为代表，以 Blog、TAG、SNS、RSS、Wiki 等社会软件的应用为核心，依据六度分隔、XML、Ajax 等新理论和技术实现的互联网新一代模式。

从 Web 1.0 到 Web 2.0 的转变，具体来说，从模式上是从读向写信息共同创造的一个改变；从基本结构上则是由网页向发表/展示工具演变；从工具上是由互联网浏览器向各类浏览器、RSS 阅读器等内容发展；从运行机制上则是由 Client Server 向 Web Services 的转变，因此，互联网内容的缔造者也由专业人士向普通用户拓展。一句话，Web 2.0 的精髓就是以人为本，提升用户使用互联网的体验。

如果说 Web 1.0 是以数据为核心的网，那么 Web 2.0 就是以报酬为出发点的互联网。看一看下面列出的最近的一些 Web 2.0 产品，读者就可以理解以上观点。

- Blog：用户织网，发表新知识以及其他用户内容链接，进而非常自然地组织这些内容。
- RSS：用户孕育发生内容自动分发、订阅。
- Podcsting：个人视频/声频的发布/订阅。
- SNS：Blog＋人以及人与人之间的链接。
- Wiki：用户共同建设一个大百科全书。

可以看到，用户在互联网上的作用越来越大，他们贡献内容、传播内容，而且提供了这些内容之间的链接以及浏览路径。在 SNS 里面，内容是以用户为核心来组织的。Web 2.0 是以用户为核心的互联网。

1.1.4 Web 的工作原理

当用户想进入万维网上的一个网页或者其他网络资源的时候，通常首先要在用户的浏览器上输入想访问网页的统一资源定位符（Uniform Resource Locator）或者通过超链接方式链接到那个网页或网络资源，之后是 URL 的服务器名部分被命名为域名系统的分布于全球的因特网数据库解析，并根据解析结果决定进入哪一个 IP（网络协议）地址（IP Address）。

接下来是为所要访问的网页向在那个 IP 地址工作的服务器发送一个 HTTP 请求。在通常情况下，HTML 文本、图片和构成该网页的一切其他文件很快会被逐一请求并发送回用户。

网络浏览器接下来的工作是把 HTML、CSS 和其他接受到的文件所描述的内容加上图像、链接和其他必须的资源显示给用户，这些构成了用户所看到的“网页”。具体访问过程如图 1-1 所示。

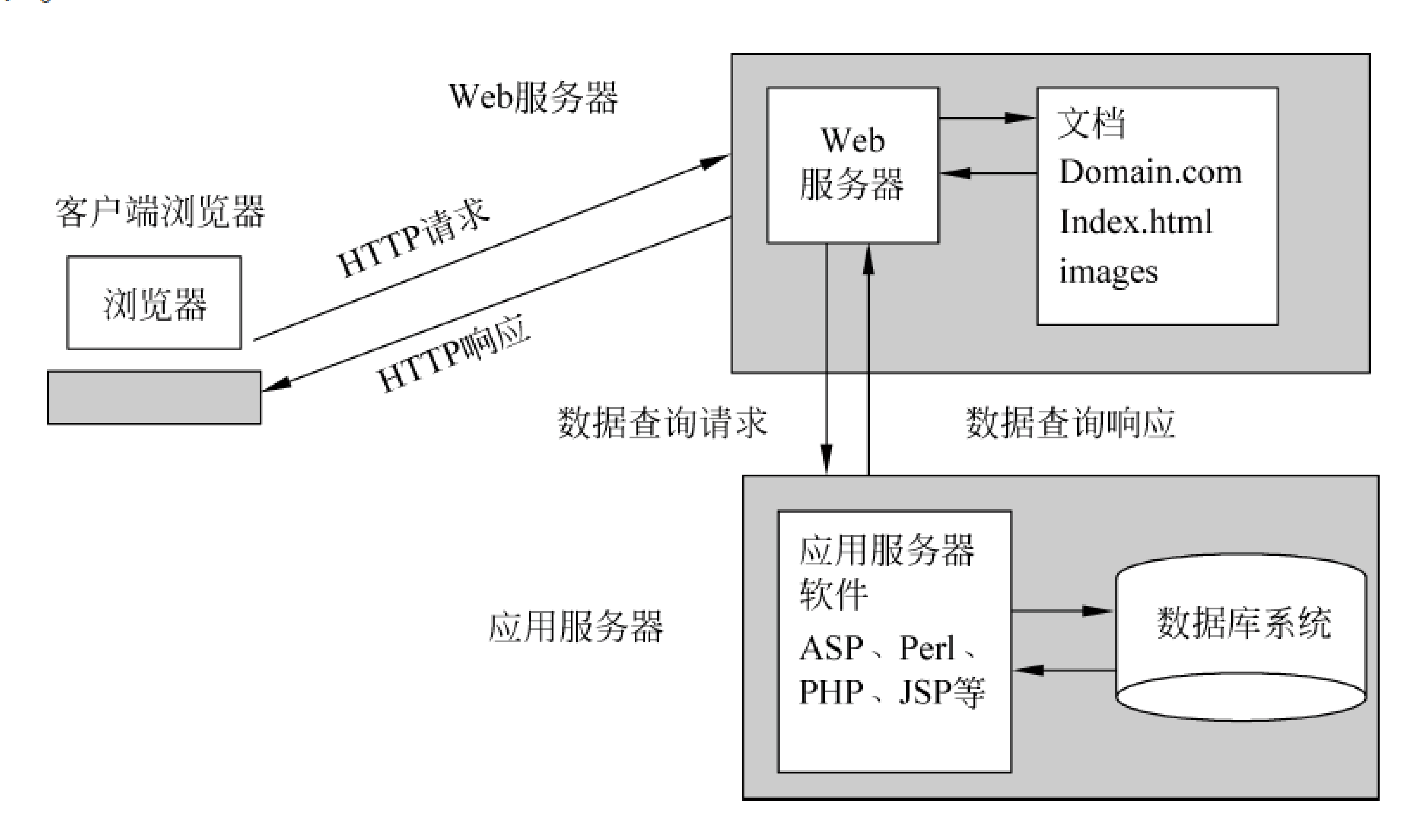


图 1-1 Web 工作原理

大多数的网页自身包含有超链接指向其他相关网页，可能还有下载、源文献、定义和其他网络资源，像这样通过超链接把有用的相关资源组织在一起就形成了一个所谓的信息的“网”，这个网在因特网上被方便地使用就构成了最早在 20 世纪 90 年代初由蒂姆·伯纳斯-

李所说的万维网。

1.2 Web 前端的由来

随着 Web 技术的不断发展，对于 Web 技术的要求越来越高，Web 开发不仅仅是网页制作这个层面，它的分工更细、包含量更大。

1.2.1 Web 前端的概念

Web 前端开发主要是指面向用户的界面或表现层开发，一般来说，Web 前端开发在绝大部分情况下都是指 JavaScript、Flash、Silverlight、CSS、HTML、Flex 等的开发设计活动，有时候也包含用 PHP、ASP. NET 等进行的动态网页开发。

1.2.2 Web 前端的开发历史和开发现状

1. Web 前端的开发历史

在 Web 发展初期，HTML 技术只能展示简单的页面，维护和更新也相当麻烦，CSS 2.0 所描述的信息结构能帮助设计师分离出表现和内容，使站点的构建和维护更加容易，因此以 CSS＋DIV 为主要技术的页面重构技术开始崭露头角，同时促进了前端开发领域的发展。

前端开发早在 Web 标准出现的时候就初见端倪，但真正有较大的发展则是在 JavaScript 逐渐成为 Web 前端开发的成熟语言之后。随着 Web 2.0 的深入人心和视频网站、SNS、网页游戏、博客、微博等相应产品陆续出现，人们对网页的各种需求不断增加，要求也越来越苛刻，因此 JavaScript 被委以重任，以 JavaScript 为标志的 Web 前端开发逐渐进入加速发展的轨道。

2. Web 前端的开发现状

到目前为止，Web 前端开发正处于发展的高峰期。由于各互联网公司都注意到站点的可用性问题，为了加强其产品的用户体验，吸引用户，各种以“用户体验”为目标的团队如雨后春笋般出现，大家只要稍微留意一下，就会发现几乎每个大的互联网公司都有属于自己的互联网团队，例如淘宝网的“淘宝 UED”、百度旗下的“百度 UFO”、腾讯的 ISD 和 CDC 等，这些团队的性质大多相似，以提高用户体验为第一目标，例如“百度 UFO”对本身团队的定位。

1.3 Web 前端开发技术

Web 前端开发技术包括 HTML、CSS、JavaScript、Ajax、jQuery 等开发语言及技术。

1. HTML

HTML（Hypertext Markup Language）超级文本标记语言是标准通用标记语言下的一

个应用,也是一种规范、一种标准。

HTML 通过标记符号来标记要显示的网页中的各个部分。网页文件本身是一种文本文件,通过在文本文件中添加标记符可以告诉浏览器如何显示其中的内容(例如文字如何处理、画面如何安排、图片如何显示等)。浏览器按顺序阅读网页文件,然后根据标记符解释和显示其标记的内容,对书写出错的标记不指出其错误,且不停止其解释执行过程,编制者只能通过显示效果来分析出错原因和出错部位。需要注意的是,对于不同的浏览器,对同一标记符可能会有不完全相同的解释,因而可能会有不同的显示效果。

2. CSS

CSS(Cascading Style Sheets)层叠样式表是一种用来表现 HTML(标准通用标记语言的一个应用)或 XML(标准通用标记语言的一个子集)等文件样式的计算机语言。

CSS 目前的最新版本为 CSS 3,它是能够真正做到网页表现与内容分离的一种样式设计语言。相对于传统 HTML 的表现而言,CSS 能够对网页中对象的位置排版进行像素级的精确控制,支持几乎所有的字体、字号样式,拥有对网页对象和模型样式编辑的能力,并能够进行初步交互设计,是目前基于文本展示最优秀的表现设计语言。CSS 能够根据不同使用者的理解能力简化或者优化写法,针对各类人群,有较强的易读性。

3. JavaScript

JavaScript 是一种直译式脚本语言,是一种动态类型、弱类型、基于原型的语言,内置支持类型。它的解释器被称为 JavaScript 引擎,为浏览器的一部分,广泛用于客户端的脚本语言,最早是在 HTML 网页上使用,用来给 HTML 网页增加动态功能。

在 1995 年,它由 Netscape 公司的 Brendan Eich 在网景导航者浏览器上首次设计而成。因为 Netscape 公司与 Sun 公司合作,Netscape 管理层希望它的外观看起来像 Java,因此取名为 JavaScript,但实际上它的语法风格与 Self 和 Scheme 较为接近。

4. Ajax

Ajax(Asynchronous JavaScript And XML,异步 JavaScript 和 XML)是一种创建交互式网页应用的网页开发技术。

Ajax =异步 JavaScript 和 XML。

Ajax 是一种用于创建快速动态网页的技术。通过在后台与服务器进行少量的数据交换,Ajax 可以使网页实现异步更新,这意味着可以在不重新加载整个网页的情况下对网页的某部分进行更新,而传统的网页(不使用 Ajax)如果需要更新内容,必须重载整个网页页面。

5. jQuery

jQuery 是继 Prototype 之后又一个优秀的 JavaScript 库,它是轻量级的 JS 库,它兼容 CSS 3,还兼容各种浏览器,jQuery 2.0 及后续版本不再支持 IE 6/7/8 浏览器。jQuery 使用户能更方便地处理 HTML、events 及实现动画效果,并且方便地为网站提供 Ajax 交互。jQuery 还有一个比较大的优势,它的文档说明很全,而且各种应用也说得很详细,同时还有

许多成熟的插件可供选择。jQuery 能够使用户的 HTML 页面保持代码和 HTML 内容分离，也就是说，不用在 HTML 里面插入一堆 JS 来调用命令，只需要定义 ID 即可。

jQuery 是一个兼容多浏览器的 JavaScript 库，核心理念是“write less，do more(写得更少，做得更多)”。jQuery 在 2006 年 1 月由美国人 John Resig 在纽约的 BarCamp 发布，吸引了来自世界各地的众多 JavaScript 高手加入，由 Dave Methvin 率领团队进行开发。如今，jQuery 已经成为最流行的 JavaScript 库，在世界前 10 000 个访问最多的网站中有超过 55%在使用 jQuery。

jQuery 是免费、开源的，使用 MIT 许可协议。jQuery 的语法设计可以使开发者更加便捷，例如操作文档对象、选择 DOM 元素、制作动画效果、事件处理、使用 Ajax 以及其他功能。除此以外，jQuery 还提供了 API 让开发者编写插件，其模块化的使用方式使开发者可以很轻松地开发出功能强大的静态或动态网页。

jQuery，顾名思义就是 JavaScript 和查询(Query)，即辅助 JavaScript 开发的库。

1.4 Web 前端开发工具

目前，常用的 Web 前端开发工具有很多，包含用于网页美工的、动画设计的、代码编辑的，下面简单地介绍几种。

1. Photoshop

Adobe Photoshop 简称 PS，它是由 Adobe Systems 开发和发行的图像处理软件。

Photoshop 主要处理以像素构成的数字图像，使用其众多的编修与绘图工具可以有效地进行图片编辑工作。PS 有很多功能，在图像、图形、文字、视频、出版等方面都有涉及。

2. Flash

Adobe Flash(原称 Macromedia Flash，简称 Flash；前身 FutureSplash)是美国 Macromedia 公司(现在已被 Adobe 公司收购)所设计的一种二维动画软件，通常包括 Adobe Flash(用于设计和编辑 Flash 文档)以及 Adobe Flash Player(用于播放 Flash 文档)。

3. Dreamweaver

Adobe Dreamweaver 简称 DW，中文名称为“梦想编织者”，它是美国 Macromedia 公司开发的集网页制作和管理网站于一身的所见即所得网页编辑器。DW 是第一套针对专业网页设计师特别发展的视觉化网页开发工具，利用它可以轻而易举地制作出跨越平台限制和跨越浏览器限制的充满动感的网页。

Adobe Dreamweaver 使用所见即所得的接口，也有 HTML 编辑功能，它有 Mac 和 Windows 系统的版本。随着 Macromedia 被 Adobe 收购，Adobe 开始计划开发 Linux 版本的 Dreamweaver。Dreamweaver 自 MX 版本开始使用了 Opera 的排版引擎 Presto 作为网页预览。

4. FrontPage

FrontPage 是 Microsoft 公司出品的一款网页制作入门级软件。FrontPage 使用方便、简单，用户会用 Word 就能制作网页，因此相对 Dreamweaver 而言 FrontPage 更容易上手。所见即所得是其特点，该软件结合了设计、程序码、预览 3 种模式。Microsoft 公司在 2006 年年底前停止提供 FrontPage 软件。

5. Notepad

Notepad 指代码编辑器或 Windows 中的“记事本”程序。它在 Windows 下主要用于文本编辑，是一款开源、小巧、免费的纯文本编辑器。在文字编辑方面，它与 Windows 写字板的功能相当。当然，更重要的是 Notepad＋＋是程序员们编写代码的利器。

6. EditPlus

EditPlus 是一款由韩国 Sangil Kim（ES-Computing）出品的小巧且功能强大的可处理文本、HTML 和程序语言的 Windows 编辑器，甚至可以通过设置用户工具将其作为 C、Java、PHP 等语言的一个简单的 IDE。

EditPlus（文字编辑器）汉化版是一套功能强大、可取代记事本的文字编辑器，拥有无限制的撤销与重做、英文拼字检查、自动换行、列数标记、搜寻取代、同时编辑多个文件、全屏幕浏览功能；而且它还有一个好用的功能，就是它有监视剪贴板的功能，同步于剪贴板，可自动粘贴进 EditPlus 的窗口中，省去粘贴的步骤；它也是一个非常好用的 HTML 编辑器，除了支持颜色标记、HTML 标记，还支持 C、C++、Perl、Java，另外，它还内建了完整的 HTML & CSS 1 指令功能，对于习惯用记事本编辑网页的朋友，它可帮助用户节省一半以上的网页制作时间，若安装 IE 3.0 以上的版本，它还会结合 IE 浏览器于 EditPlus 窗口中，让用户可以直接预览编辑好的网页（若没安装 IE，也可指定浏览器路径）。因此，它是一个相当棒又多用途、多状态的编辑软件。

1.5　Web 前端工程师的职业要求

一位好的 Web 前端开发工程师在知识体系上既要有广度，又要有深度，所以很多大公司即使出高薪也很难招聘到理想的前端开发工程师。以前会 Photoshop 和 Dreamweaver 就可以制作网页，现在只掌握这些已经远远不够了。无论是在开发难度上，还是在开发方式上，现在的网页制作都更接近传统的网站后台开发，所以现在不再叫网页制作，而是叫 Web 前端开发。Web 前端开发在产品开发环节中的作用变得越来越重要，而且需要专业的前端工程师才能做好，这方面的专业人才近两年来备受青睐。Web 前端开发是一项很特殊的工作，涵盖的知识面非常广，既有具体的技术，又有抽象的理念。简单地说，它的主要职能就是把网站的界面更好地呈现给用户。

如何才能做得更好呢？

第一，必须掌握基本的 Web 前端开发技术，其中包括 CSS、HTML、DOM、BOM、Ajax、JavaScript 等，在掌握这些技术的同时，还要清楚地了解它们在不同浏览器上的兼容情况、

渲染原理和存在的 Bug。

第二，在一名合格的前端工程师的知识结构中，网站性能优化、SEO 和服务器端的基础知识也是必须掌握的。

第三，必须学会运用各种工具进行辅助开发。

第四，除了要掌握技术层面的知识外，还要掌握理论层面的知识，包括代码的可维护性、组件的易用性、分层语义模板和浏览器分级支持等。

可见，看似简单的网页制作如果要做得更好、更专业，真的是不简单，这就是前端开发的特点，也是让很多人困惑的原因。如此繁杂的知识体系让新手学习起来无从下手，对于老手来说，也时常不知道下一步该学什么。

代码质量是前端开发中应该重点考虑的问题之一。例如，实现一个网站界面可能会有无数种方案，但有些方案的维护成本会比较高，有些方案会存在性能问题，而有些方案更易于维护，并且性能也比较好，这里的关键影响因素就是代码质量。CSS、HTML、JavaScript 这 3 种前端开发语言的特点是不同的，对代码质量的要求也不同，但它们之间又有着千丝万缕的联系。

第2章 Photoshop CS6概述

本章学习目标：

✍了解 Photoshop 的发展及应用。
✍掌握位图与矢量图的特点及区别。
✍掌握 Photoshop 的窗口组成。
✍掌握 Photoshop 文件的基本操作。

2.1 Photoshop 简介

Adobe Photoshop CS6 是目前应用最广泛的平面设计软件之一，它从 Photoshop 1.0 到目前的 Photoshop CS6 共经历了 13 个版本，功能不断增强，应用越来越广，在图像、图形、文字、视频、出版等方面都有涉及。

2.1.1 Photoshop 的起源

Photoshop 的主要设计师 Thomas Knoll 的父亲 Glenn Knoll 是密歇根大学教授，同时也是一个摄影爱好者，他家的地下室是一个暗房，他的两个儿子 Thomas 和 John 从小就跟着爸爸玩暗房，但 John 似乎对当时刚刚开始发行的个人电脑更感兴趣，此后 Thomas 也迷上个人电脑，并在 1987 年买了一台苹果电脑（MacPlus）用来帮助完成他的博士论文。Thomas 发现：当时的苹果电脑无法显示带灰度的黑白图像，因此他自己写了一个程序 Display；而 John 这时在星球大战导演 Lucas 的电影特殊效果制作公司 Industry Light Magic 工作，对 Thomas 的程序很感兴趣。兄弟俩在此后的一年多把 Display 不断修改为功能更为强大的图像编辑程序，经过多次改名后，在一个展会上他们接受一个参展观众的建议，把程序改名为 Photoshop。此时的 Display/Photoshop 已经有 Level、色彩平衡、饱和度等调整。此外，John 还写了一些程序，后来成为插件（Plug-in）的基础。他们的第一个商业成功是把 Photoshop 交给一个扫描仪公司搭配卖，名字叫 Barneyscan XP，版本是 0.87。与此同时，John 继续找其他买家，包括 SuperMac 和 Aldus 都没有成功。最终他们找到了 Adobe 公司的 Russell Brown（Adobe 公司的艺术总监），Russell Brown 此时已经在研究是否考虑另外一家公司 Letraset 的 ColorStudio 图像编辑程序，看过 Photoshop 以后他认为 Knoll 兄弟的程序更有前途，在 1988 年 7 月他们口头决定合作，真正的法律合同到次年 4 月才完成。Adobe 公司决定保留 Photoshop 这个名字。

1990 年 2 月，Photoshop 1.0 版本发行，它给计算机图像处理行业市场带来巨大的冲击。除了拥有其他软件没有的特点外，它还获得了天时，当时正值计算机桌面革命炒得火热，桌面的发展更为它创造了有利条件。之后，Photoshop 随着功能不断强大，版本也在升级，到 2003 年，Adobe 的 Creative Suite 套装将 Adobe Photoshop 8 更名为 Adobe Photoshop CS。Adobe Photoshop 有两个发行版本，即标准版 Adobe Photoshop 和扩展版 Adobe Photoshop Extended，扩展版除包含标准版的所有功能之外，还增加了 3D 处理功能、动画图形编辑功能和高级图像分析功能。

2012 年，Adobe Photoshop CS6 正式发布，在这个版本中整合了其 Adobe 专有的 Mercury 图像引擎，并通过显卡核心 GPU 提供了强悍的图片编辑能力。Content-Aware Patch 能帮助用户更加轻松、方便地选取区域，方便用户抠图等操作。

2.1.2 Photoshop 的应用

Photoshop 的应用领域大致包括数码照片处理、广告摄影、视觉创意、平面设计、艺术文字、建筑效果图后期修饰及网页制作等，下面分别进行介绍。

(1) 数码照片处理：在 Photoshop 中可以进行各种数码照片的合成、修复和上色操作，例如为数码照片更换背景、为人物更换发型、去除斑点、数码照片的偏色校正等。Photoshop 同时也是婚纱影楼设计师们的得力助手。

(2) 广告摄影：广告摄影作为一种对视觉要求非常严格的工作，要用最简洁的图像和文字给人以最强烈的视觉冲击，其最终作品往往要经过 Photoshop 的艺术处理才能得到满意的效果。

(3) 视觉创意：视觉创意是 Photoshop 的特长，通过 Photoshop 的艺术处理可以将原本不相干的图像组合在一起，用户也可以发挥想象自行设计富有新意的作品，利用色彩效果等在视觉上表现全新的创意。

(4) 平面设计：平面设计是 Photoshop 应用最为广泛的领域，无论是图书封面，还是招贴、海报，这些具有丰富图像的平面印刷品基本上都需要使用 Photoshop 软件对图像进行处理。

(5) 艺术文字：普通的艺术文字经过 Photoshop 的艺术处理会变得精美绝伦，利用 Photoshop 可以使文字发生各种各样的变化，并且利用这些艺术化处理后的文字也可以为图像增加效果。

(6) 建筑效果图后期修饰：当在制作的建筑效果图中包括许多三维场景时，人物与配景包括场景的颜色常常需要在 Photoshop 中增加并进行调整。

(7) 网页制作：网络的迅速普及是促使更多的人学习和掌握 Photoshop 的一个重要原因，因为在制作网页时 Photoshop 是必不可少的网页图像处理软件，而且发挥的作用越来越大。

2.1.3 图像的基本概念

1. 像素和分辨率

在 Photoshop 中，像素(pixel)是组成图像的最基本单元，它是一个小矩形颜色块。一

个图像通常由很多像素组成，这些像素被排成横行或纵列，当用缩放工具把图像放大到一定比例时，就可以看到类似马赛克的效果。每个像素都有不同的颜色值，单位长度的像素越多，分辨率(ppi)越高，图像的品质就越好。图 2-1 所示为显示器上以正常比例显示的图像，当把图像放大到一定的比例后，就会看到图 2-2 所示的类似马赛克的效果。



图 2-1 显示器上正常显示的图像

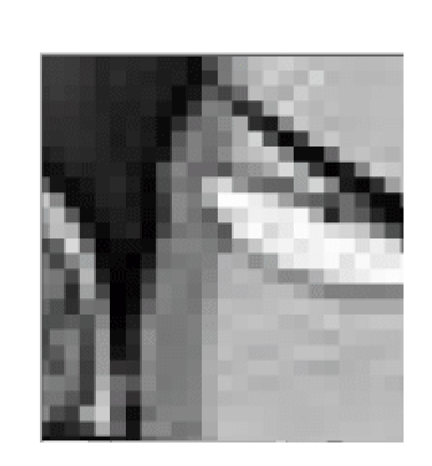


图 2-2 图像放大后的马赛克效果

图像分辨率是指图像在一个单位打印长度内像素的个数，分辨率的单位是 ppi(pixels per inch)。例如，图像分辨率是 72ppi，即在每英寸长度内包含 72 个像素，也就是在每一平方英寸的图像中有 5184 个像素(72×72)。图像分辨率越高，输出效果越清晰。

分辨率的高低和图像大小之间有着密切的关系，分辨率越高，所包含的像素越多，图像的信息量越大，因此文件也就越大。此外，图像的清晰度也与像素的总数有关，如果像素固定，那么提高分辨率虽然可以使图像变得比较清晰，但尺寸却会变小；反之，降低分辨率图像会变大，但画面质量会变得比较粗糙。像素数目和分辨率共同决定了打印时图像的大小，像素相同但分辨率不同的图像，打印时的大小也不相同。

另外，大家经常提到的输出分辨率是以 dpi(dots per inch，每英寸所含的点)为单位的，它是针对输出设备而言的，通常激光打印机的输出分辨率为 300～600dpi，照排机要达到 1200～2400dpi 或更高。

2. 点阵图和矢量图

点阵图是由 Adobe Photoshop、Paint 等软件产生的，如果将此类图放大到一定程度，就会发现它是由一个个小方格组成的，这些小方格被称为像素，且每个像素都有一个明确的颜色，故此类图又被称为像素图。在整张图中，单位面积内所包含的像素越多越能表现出图片细微的部分。其中，分辨率和点阵图有着密不可分的关系，分辨率越高，单位面积内的像素就越多，图像也就越清晰；反之，分辨率太低，或将图片的显示比例放得过大，就会造成图像变模糊且产生锯齿边缘和色调不连续的情况。

由于点阵图是由一连串排列的像素组合而成的，它并不是独立的图形对象，所以不能编辑图像中的个别对象。如果要编辑其中部分区域的图像，就必须首先精确地选取需要编辑的像素范围，然后再进行编辑，能够处理点阵图的软件有 Photoshop、Photo Impact、Painter 和 CorelDRAW 软件内的 CorelPhotoPaint 等。

矢量图是由经过精确定义的直线和曲线组成的，这些直线和曲线称为向量，因此矢量图又称为向量图。其中每一个对象都是独立的个体，它们都有各自的色彩、形状、尺寸和位置坐标等属性。在矢量图编辑软件中可以任意改变图中某个对象的属性，而不会影响到其他的对象，也不会降低图形的品质。

矢量图与分辨率无关，也就是说，可以将它们缩放到任意尺寸，可以按任意分辨率打印，且不会丢失细节或降低清晰度，因此矢量图最适合表现醒目的图形，这种图形无论缩放到何种程度均能保持线条清晰，不会失真。

矢量图的文件大小只与图形的复杂程度有关，一般需要的存储空间很小，在绘制与编辑时对计算机的内存要求较低；在输出时，可以用打印机或印刷机等输出设备的最高分辨率进行打印。矢量图一般是直接在计算机上绘制而成的，可以绘制、编辑矢量图的软件有Illustrator、CorelDRAW、Freehand 和 Expression 等。

3. 颜色深度

颜色深度(Color Depth)用来度量图像中有多少种颜色信息可用于显示或打印像素，其单位是位(bit)，所以颜色深度有时也称为位深度或像素深度。常用的颜色深度是 1 位、8 位、24 位和 32 位，颜色深度为 1 位的像素有两个可能的数值，即 0 或 1。较大的颜色深度(每像素信息的位数更多)意味着数字图像具有较多的可用颜色和较精确的颜色表示。

因为一个 1 位的图像包含两种颜色，所以 1 位的图像最多可由两种颜色组成，每个像素的颜色只能是黑色或白色；一个 8 位的图像包含 2^8 种颜色，或 256 级灰阶，每个像素可能是 256 种颜色中的任意一种；一个 24 位的图像包含 1670 万(2^{24})种颜色；一个 32 位的图像包含 2^{32} 种颜色，但很少这样讲，这是因为 32 位的图像可能是一个具有 Alpha 通道的 24 位图像，也可能是 CMYK 色彩模式的图像，这两种情况下的图像都包含有 4 个 8 位的通道。图像色彩模式和色彩深度是相关联的(一个 RGB 图像和一个 CMYK 图像都可以是 32 位)，但不总是这种情况。Photoshop 也支持 16 位/通道，可产生 16 位灰度模式的图像、48 位 RGB 模式的图像、64 位 CMYK 模式的图像，表 2-1 列出了常见的色彩深度、颜色数量和色彩模式的关系。

表 2-1　色彩深度、颜色数量和色彩模式的关系

色彩深度	颜色数量	色彩模式
1 位	2(黑和白)	位图
8 位	256	索引颜色/灰度
16 位	65 536	灰度 16 位/通道
24 位	1670 万	RGB
32 位		CMYK、RGB
48 位		RGB 16 位/通道

2.2 Photoshop CS6 界面简介

正确安装 Photoshop 后，单击 Windows 桌面任务栏上的【开始】按钮，在弹出的【开始】菜单中选择【所有程序】| Adobe Photoshop CS6，即可启动 Photoshop 软件。在 Adobe Photoshop CS6 中，默认窗口界面是黑色的。

启动 Photoshop CS6 后，可以看出 Photoshop CS6 的界面仍然延续了 Adobe 公司产品的一贯风格，如图 2-3 所示。按照功能，Photoshop CS6 的界面具体分为标题栏、菜单栏、工

具箱、工具选项栏、面板区、图像窗口、状态栏等,下面分别进行介绍。

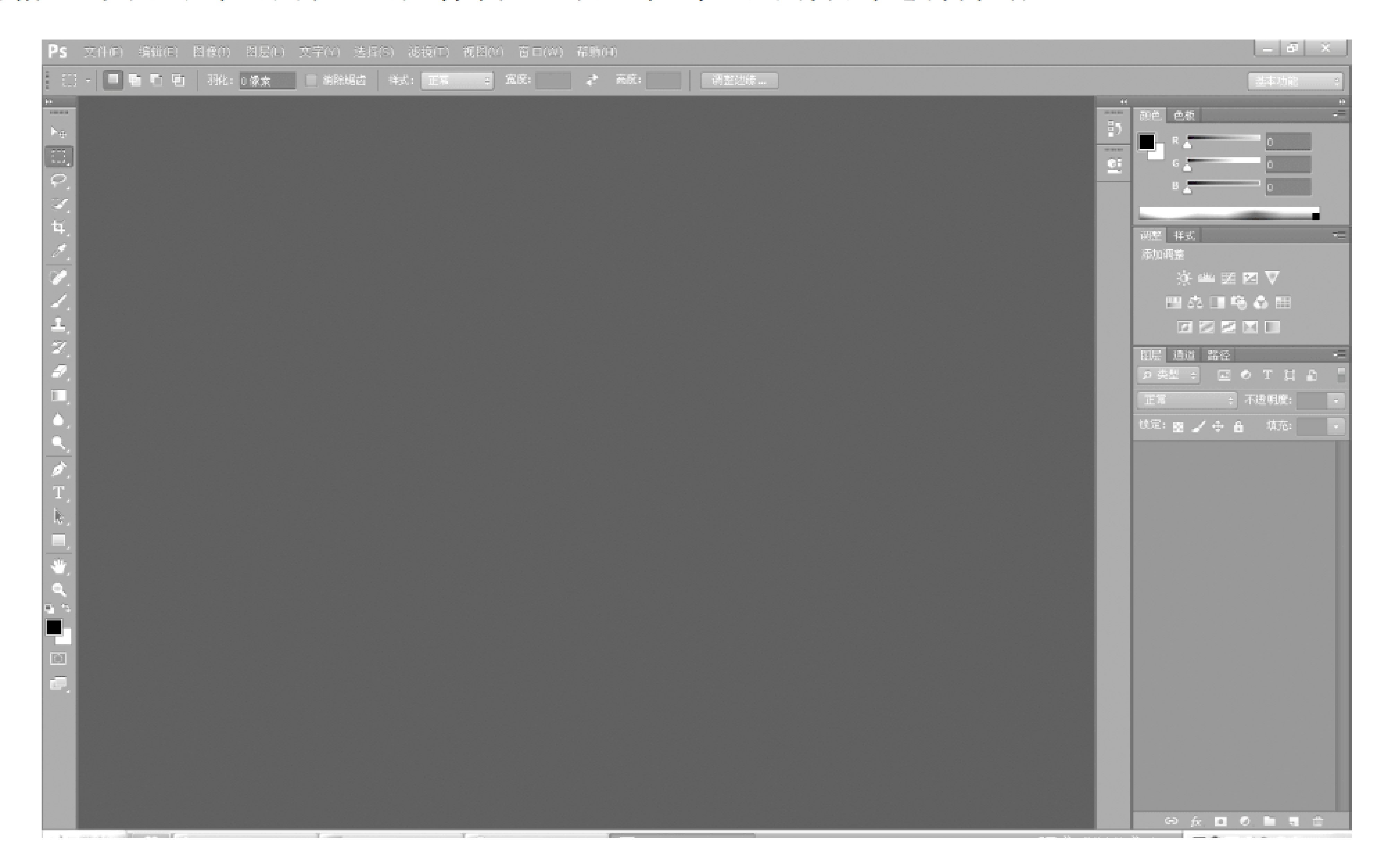


图 2-3 窗口界面介绍

1. 标题栏

标题栏位于界面最上方,其左侧显示的是软件图标及名称,右侧是用来控制界面显示状态的 3 个按钮,从左至右分别为【最小化】按钮、【最大化/还原】按钮和【关闭】按钮。

2. 菜单栏

菜单栏位于标题栏下方,包含了用于图像处理的各类命令,共有【文件】、【编辑】、【图像】、【图层】、【文字】、【选择】、【滤镜】、【视图】、【窗口】和【帮助】10 个菜单,每一个菜单下又有若干个菜单命令,选择相关的菜单命令可以执行相应的操作。

在下拉菜单中有些命令的后面有省略号,表示选择此命令可以弹出相应的对话框;有些命令的后面有向右的黑色三角形,表示此命令还有下一级菜单;还有一部分命令显示为灰色,表示该命令当前不能使用,只有在满足一定的条件之后才可使用。

3. 工具箱

工具箱默认位于界面左侧,通过单击工具箱上部的双箭头可以使工具箱中的工具排列在单列和双列间进行转换。工具箱中包含了用于图像处理和图形绘制的各种工具。

如果要查看工具的名称,可将鼠标指针移至该工具处,稍等片刻,系统将自动显示工具名称的提示。工具箱中有些工具按钮的右下角带有黑色的小三角符号,表示该工具还隐藏有其他同类工具,将鼠标指针移至此类按钮上按下鼠标左键不放,隐藏工具即会显示出来。在隐藏的工具组中选择所需的工具,则该工具将成为当前工具。

4. 工具选项栏

工具选项栏(以下简称选项栏)位于菜单栏上方,其功能是显示工具箱中当前被选择工具的相关参数和选项,以便对其进行具体设置。

5. 面板区

面板区默认位于界面右侧,主要用于存放 Photoshop CS6 提供的功能面板(以下简称面板)。Photoshop CS6 共提供了 28 种面板,利用这些面板可以对图层、通道、工具、色彩等进行设置和调整,用户可以利用菜单栏中的【窗口】命令显示和隐藏面板。

6. 图像窗口

在 Photoshop CS6 中,图像窗口默认以选项卡的方式显示所打开的图像文件,拖曳可以使图像窗口浮动。在浮动的图像窗口中,最上方的标题栏中显示图像的相关信息,例如图像的文件名称、文件类型、显示比例、目前所在图层以及所使用的颜色模式和位深度等。

7. 状态栏

状态栏位于图像窗口的最下方,显示当前图像的状态及操作命令的相关提示信息。其中,最左侧的数值显示当前图像的百分比,用户可以通过直接修改这个数值来改变图像的显示比例;显示百分比的右侧是当前图像文件的信息,单击文件信息右侧的按钮会弹出扩展菜单,如图 2-4 所示,可以改变对当前文档的设置。

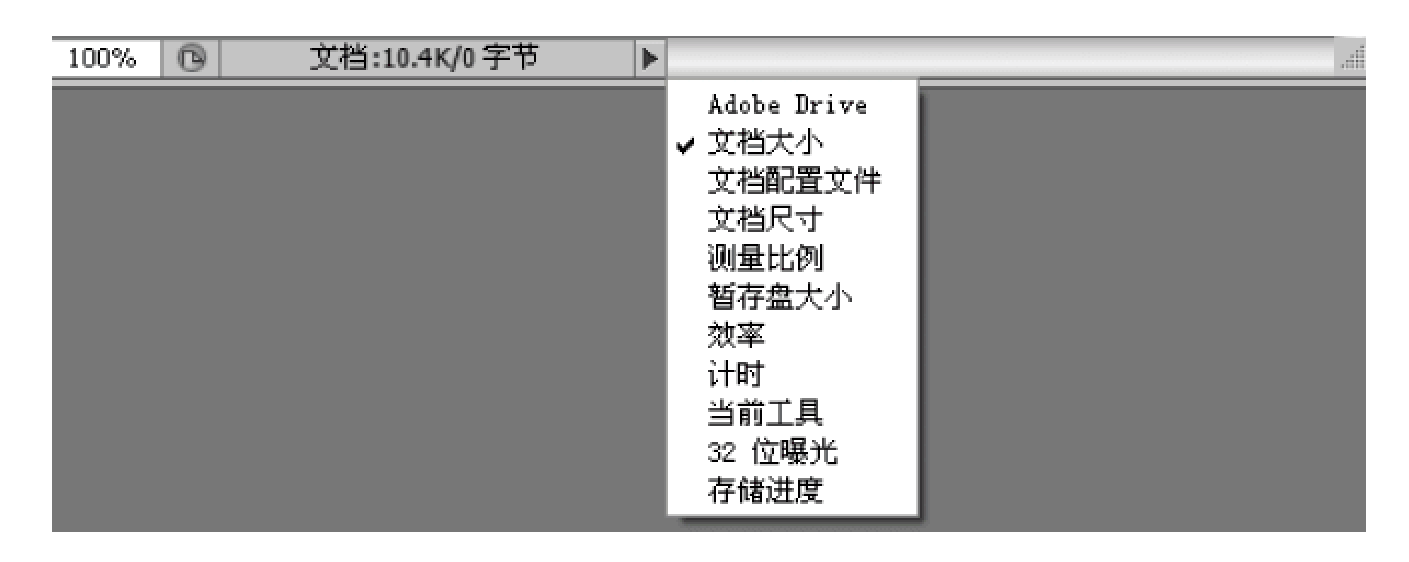


图 2-4 状态栏菜单

以上介绍的是 Photoshop CS6 的默认界面,为了操作方便,用户可以对界面中各部分的位置进行调整,当需要时可以将工具箱、选项栏和面板区进行隐藏。

将鼠标指针移到工具箱、选项栏、面板区或图像窗口最上方的标题栏上,拖曳鼠标可以移动它们的位置。选择【窗口】菜单,在弹出的下拉菜单中选择或取消相应的面板,可以分别将各面板显示或隐藏。按 Tab 键,可以将工具箱、选项栏和所有面板同时显示或隐藏;在按住 Shift 键的同时按 Tab 键,可以将界面窗口中的所有面板同时显示或隐藏。选择菜单栏中的【窗口】|【工作区】|【复位基本功能】命令,可以使界面恢复到初始默认状态。



2.3 Photoshop CS6 的基本操作

下面介绍 Photoshop 文件的建立及保存、文件的显示等基础操作。

2.3.1 文件的新建

选择菜单栏中的【文件】|【新建】命令，在弹出的【新建】对话框中进行参数设置，然后单击【确定】按钮完成文件的新建，如图 2-5 所示。



图 2-5 【新建】对话框

在【新建】对话框中可对所建文件进行如下设定：

- 在【名称】文本框中可输入图像名称。
- 在【预设】下拉列表中可选择一些内定的图像尺寸。
- 在【宽度】、【高度】和【分辨率】文本框中可输入自定的尺寸和分辨率，在文本框右侧的下拉列表中还可以选择不同的度量单位。分辨率的单位一般采用“像素/英寸”，如果制作的图像用于印刷，需设定“300 像素/英寸”的分辨率。
- 在【颜色模式】下拉列表中可设定图像的色彩模式。
- 在【背景内容】下拉列表中可选择新建图像的背景颜色，包括“白色”、“背景色”和“透明”3 种类型。

2.3.2 文件的存储

文件的存储命令主要包括【存储】和【存储为】两种。在对新建的文件编辑后进行存储，使用这两种命令操作是一样的，都是为当前文件命名并进行保存；但若对打开的文件进行编辑后保存，就要区分【存储】和【存储为】命令，前者是将文件以原文件名进行保存，而后者是将修改后的文件重新命名进行保存，如图 2-6 所示。



图 2-6　文件的存储

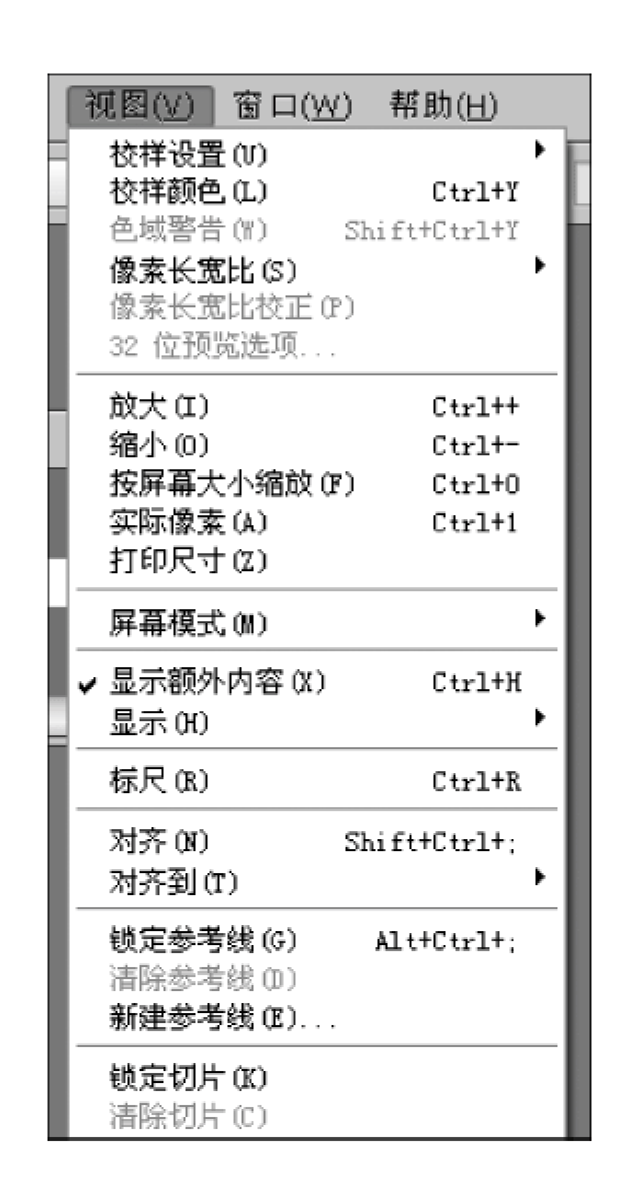


图 2-7　缩放命令

2.3.3　图像的浏览

在 Photoshop 的【视图】菜单下有很多命令用来控制图像的不同显示比例，如图 2-7 所示。一个图像的最大显示比例是 3200%，最小则显示一个像素。注意，使用这些命令只是放大或缩小了图像的显示比例，并没有真正地改变图像的尺寸，用户也可使用相应的快捷键完成图像的缩放。

1.【放大】与【缩小】命令

选择【视图】菜单下的【放大】与【缩小】命令，可以改变当前图像的显示比例。每使用一次【放大】或【缩小】命令，图像的显示尺寸就会在原来的基础上放大一倍或缩小为原来的 1/2。

2.【按屏幕大小缩放】

选择【视图】菜单下的【按屏幕大小缩放】命令，或双击工具箱中的抓手工具图标，可以按屏幕大小显示当前图像的最大比例。

全屏显示会受到工具箱和面板的限制，当工具箱和面板以默认位置分布在屏幕两侧时，全屏显示会自动让出屏幕两侧的位置，而以一个较小的图像窗口来显示整幅图像，只有当关闭或隐藏所有的工具箱和面板时才能真正在屏幕上实现全屏显示。

3.【实际像素】

【实际像素】表示以一个显示器的屏幕像素对应一个图像像素时的显示比例，即 100%

的显示比例。直接选择【视图】菜单下的【实际像素】命令，或单击工具箱中的缩放工具图标，便可以按100%的实际像素显示。

4.【打印尺寸】

使用【视图】菜单下的【打印尺寸】命令可以在屏幕上显示出图像的实际打印大小。

实际打印尺寸不考虑图像的分辨率，而是以图像本身的宽度和高度(打印时的尺寸)来表示一幅图像的大小。

5. 缩放工具

使用缩放工具可以将图像成比例地放大或缩小，以便于用户对图像进行观察和修改。

在选择缩放工具时，鼠标指针在图像窗口内显示为一个带加号的放大镜，单击即可实现图像的成倍放大。在按住 Alt 键使用放大工具时，鼠标指针变为一个带减号的缩小镜，单击可将图像按50%缩小。

6. 抓手工具

Photoshop 工作区的范围是有限的，当需要对图像的局部进行精细处理时，有时需要将图像放大显示到超出图像窗口的范围，此时图像在窗口内无法完全显示，利用工具箱中的抓手工具可以在窗口中移动图像对其进行局部观察和修改。

7.【导航器】面板

【导航器】面板是用来观察图像的，可方便用户对图像进行缩放，如图 2-8 所示。在面板的左下角显示百分比数字，用户可直接输入值，按 Enter 键确认后，图像即会按照相应的百分比显示，在导航器中会有相应的预览图；也可以用鼠标拖曳导航器右下方的三角滑块来改变缩放比例。单击左侧较小的图标可以使图像缩小显示，单击右侧较大的图标可以放大显示。



图 2-8 【导航器】面板

2.3.4 面板的显示与隐藏

在【窗口】菜单中有一部分面板命令的左侧显示有对号，说明此面板目前在工作区中处于显示状态；没有显示对号的面板命令，说明相应面板已被关闭，处于隐藏状态，如图 2-9 所示。

选择【窗口】菜单中的某一面板命令，使其左侧显示对号，表示显示相应面板；再次选择，使其左侧不显示对号，表示隐藏相应面板。

2.3.5 标尺、网格和参考线

标尺、网格和参考线是 Photoshop CS6 中的帮助工具，使用的频率非常高。用户使用这3种工具可以给图形绘制和图像处理带来极大的方便，可以帮助用户精确地定位和对齐对象。

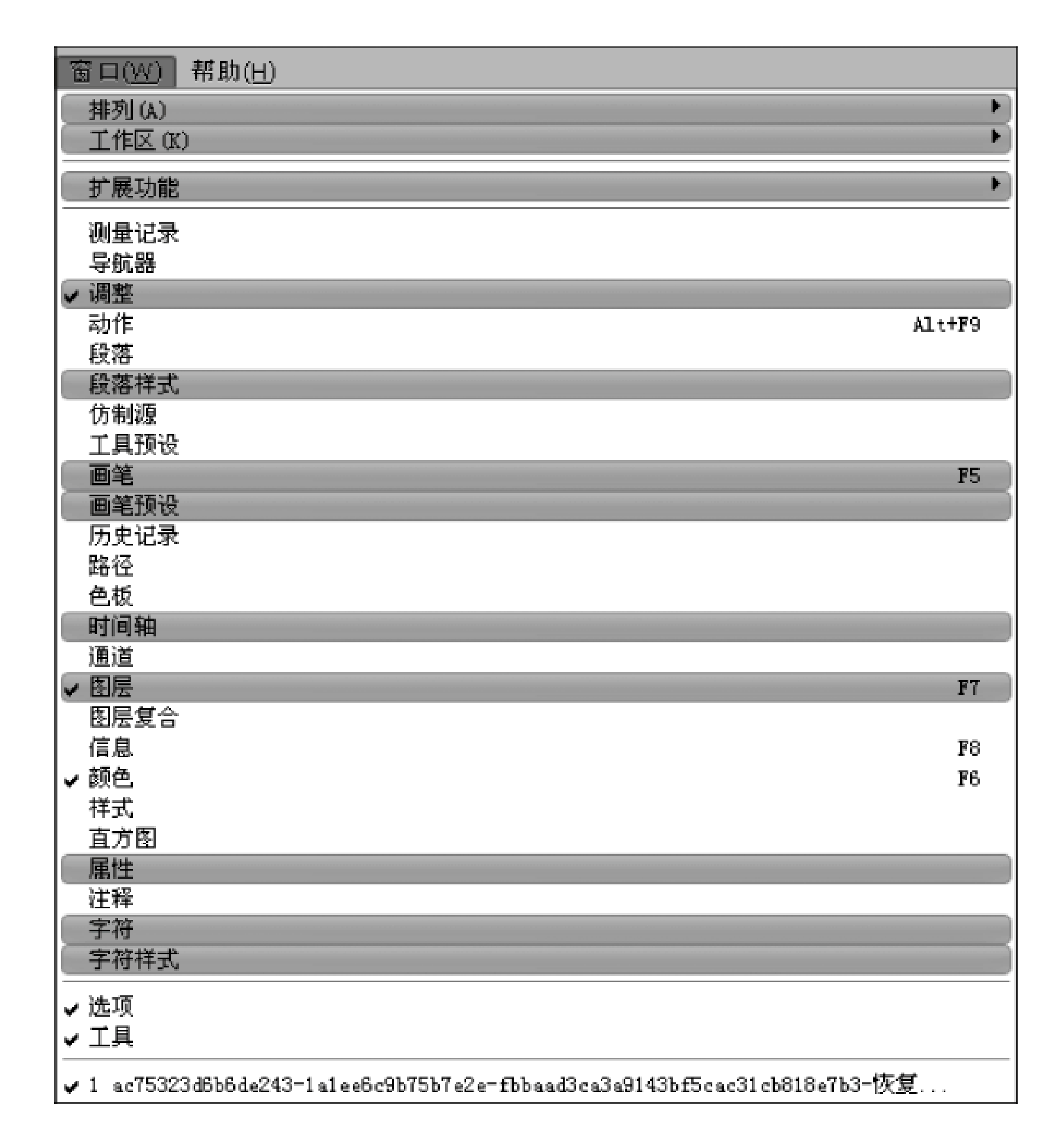


图 2-9 【窗口】菜单

1. 标尺

选择菜单栏中的【视图】|【标尺】命令，在图像窗口的左边和上边会显示标尺；当再次选择此命令时，可以将标尺隐藏，如图 2-10 所示。

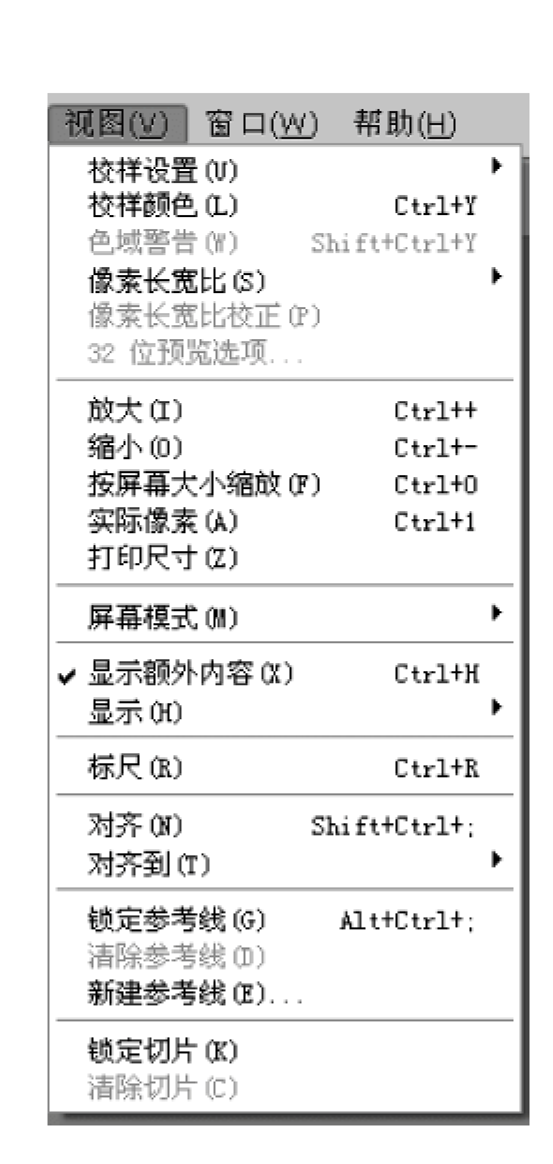


图 2-10 【视图】菜单

2. 网格

选择菜单栏中的【视图】|【显示】|【网格】命令，在当前文件的图像窗口中会显示出网格；当再次选择该命令时，可以将网格隐藏。

3. 参考线

将标尺显示在视图窗口中，将鼠标指针放在标尺的位置，按住鼠标左键向外拖曳，即可添加参考线。

选择菜单栏中的【视图】|【新建参考线】命令，可弹出【新建参考线】对话框，在此对话框中设置参考线的方向，并直接以输入数值的方式确定参考线的位置。

选择菜单栏中的【视图】|【锁定参考线】命令，可将所有的参考线锁定。当拖曳参考线到图像窗口之外时释放鼠标左键，即可将其删除。选择菜单栏中的【视图】|【清除参考线】命令，可以将图像窗口中的所有参考线全部删除。

2.4 综合应用

2.4.1 界面颜色的修改

Photoshop CS6 默认窗口界面的颜色为黑色，如果想调整界面颜色，则需要选择菜单栏中的【编辑】|【首选项】|【界面】命令，打开【界面】对话框，如图 2-11 所示。在【外观】中选择用户想应用的颜色，如果以前用过 Photoshop 的其他版本，一般会选择浅灰色。



图 2-11 【界面】对话框

2.4.2 网格线的设置

如果想对网格间隔、颜色进行更改，需要选择菜单栏中的【编辑】|【首选项】|【参考线、网格和切片】命令，打开【参考线、网格和切片】对话框进行相应的设置，如图 2-12 所示。



图 2-12 【参考线、网格和切片】对话框

2.4.3 【首选项】命令

选择菜单栏中的【编辑】|【首选项】命令,在打开的级联菜单中可以对 Photoshop 软件进行初始化设置,包括对内存、暂存盘、历史记录、标尺、光标等进行初始化设置,如图 2-13 所示。

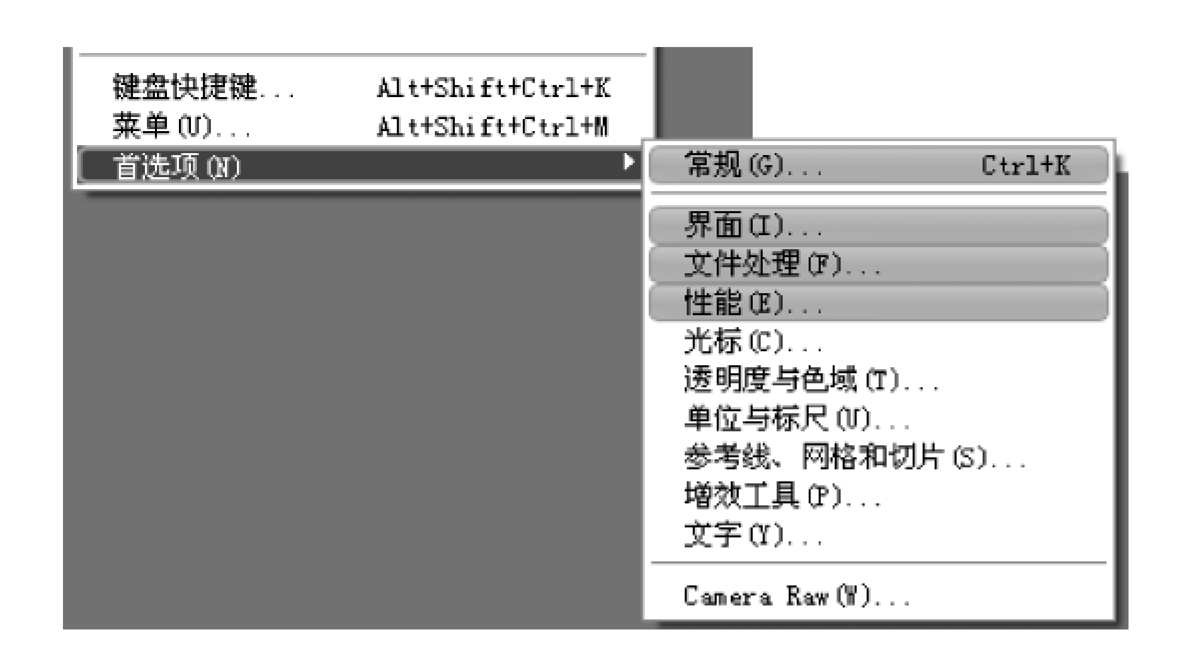


图 2-13 【首选项】命令

2.4.4 显示与隐藏浮动面板

在图 2-14 所示的【窗口】菜单中选择任意面板命令，则将此面板显示，如图 2-15 所示，其面板命令前有对号显示；再次选择该命令，对号消失，则该面板隐藏。



图 2-14 【窗口】菜单



图 2-15 显示的浮动面板

第3章 Photoshop工具的应用

本章学习目标：

✍了解 Photoshop 工具箱中的各种工具。

✍掌握 Photoshop 工具箱中常用工具的使用。

3.1 选区工具

在 Photoshop 中所做的操作都是对选区有效的，选区在 Photoshop 中表现为由滚动的虚线（俗称蚂蚁线）围成的区域。

3.1.1 选框工具

选框工具包括矩形选框工具、椭圆选框工具、单行选框工具、单列选框工具。在工具箱中按住选框工具不放，将弹出图 3-1 所示的工具列表。

在列表中选择要使用的工具，然后在图像中拖动鼠标，会出现一个虚线框，我们称之为选框，选框内的部分就是选区，后续进行的所有操作只对选区内的图像起作用。

图 3-1 选框工具列表

1. 选框工具选项栏

在使用选框工具时，其选项栏如图 3-2 所示。



图 3-2 选框工具选项栏

- 单击【新选区】按钮在图像中创建选区，新的选区将代替原来的选区。
- 单击【添加到选区】按钮在图像中创建选区，新创建的选区与原来的选区合并为新的选区。
- 单击【从选区减去】按钮，如果新创建的选区与原来的选区有相交部分，则从原选区中减去相交部分。
- 单击【与选区交叉】按钮，如果新创建的选区与原来的选区有相交部分，则保留相交选区。

- 羽化：决定选区边缘的柔化程度，可以在数值框内输入羽化的数值，其取值范围为0～255。
- 消除锯齿：勾选此复选框，可以使边缘看起来柔和，达到抗锯齿的目的。
- 样式：此下拉列表中有3个选项，选择【正常】选项，可以在图像中创建任意大小与比例的选区；选择【固定比例】选项，可以设置将要创建选区的宽度与高度的比例；选择【固定大小】选项，可以设置将要创建选区的宽度和高度值。
- 【调整边缘】：可以重新设置选区的边缘效果。

2. 选区的基本操作

如果当前图像中没有选区，选择工具箱中的选框工具建立特殊选区，基本操作方法如下：

- 按住 Shift 键不放，在图像中拖曳鼠标指针，可以在图像中创建正方形或圆形选区。
- 按住 Ctrl 键不放，在图像中拖曳鼠标指针，可以生成一个以鼠标指针落点为中心的选区。

取消选区的常用方法有两种，一种是选择菜单栏中的【选择】|【取消选择】命令，另一种是按 Ctrl+D 键。

3.1.2 移动工具

如果想调整各图层图像间的相对位置，必须要用到工具箱中的移动工具。移动工具主要用于将某些特定的图像进行移动、复制，这一操作可以在同一幅图像中进行，也可以在不同的图像间进行。利用该工具还可以方便地对链接图层进行对齐、平均分布以及对图像进行变形等操作。

在工具箱中选择移动工具后，其选项栏如图 3-3 所示。

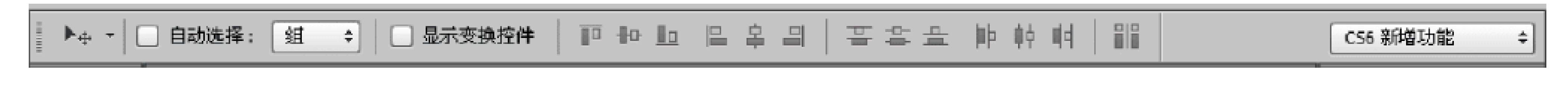


图 3-3 移动工具选项栏

移动工具选项栏中的选项有三部分功能，即自动选择要移动的图层或组、自由变形、对齐和分布图层。

1. 自动选择图层

【自动选择】下拉列表框中有【图层】和【组】两个选项。勾选【自动选择】复选框并在下拉列表框中选择【图层】选项，则可以利用移动工具在图像中单击自动选择鼠标指针所在位置上第一个有可见像素的图层并进行变换；如勾选【自动选择】复选框并在下拉列表框中选择【组】选项，则可以通过单击选择成组图层中的某一个图层中的像素来选择成组图层，在变换时会对该成组图层中的所有图层都产生作用。

2. 利用移动工具变形

利用移动工具可以对一些图像的大小和角度进行调整，即对图像进行变形修改。

1）利用移动工具对图像进行自由变形

在图像窗口中选择要进行变形的图像，或者当前图层不是背景层时选择移动工具，在选项栏中勾选【显示变换控件】复选框，图像就会出现变换控件框，变换控件框的4条边上的小矩形称为调节点，虚连线称为边线，中间的控点称为参考点，如图3-4所示。将鼠标指针移动到变换控件框的调节点或边线上单击，当变换控件框变为实线框时，可以对框内的图像进行变形修改。



图 3-4　变形控件框



（1）设置参考点的位置：

参考点是变形的基准，在图像窗口中直接拖动参考点可以调整参考点的位置。

（2）缩放：

① 将鼠标指针移至变换控件框的调节点上，拖动鼠标可以对图像进行任意缩放变形。

② 将鼠标指针移至变换控件框的边线上，拖动鼠标可以对图像进行水平或垂直缩放变形。

③ 按住 Alt 键，将鼠标指针移至变换控件框的调节点上拖动，可以对图像以参考点为基准进行对称缩放。

④ 按住 Shift 键，将鼠标指针移至变换控件框的4个角的调节点上拖动，可以对图像进行等比例缩放。

（3）旋转：

① 将鼠标指针移至变换控件框的边线或调节点上，拖动鼠标可以对图像以参考点为中心进行旋转。

② 按住 Shift 键，将鼠标指针移至变换控件框的边线上，拖动鼠标可以对图像以参考点为中心，按15°的增量进行旋转。

（4）斜切（调节点只能在水平或垂直方向上移动）：

按住 Ctrl＋Shift 键，将鼠标指针移动至变换控件框的调节点上，拖动可以对图像进行拉伸；将鼠标指针移动至变换控件框的边线上，拖动鼠标可以对图像进行倾斜变形。

（5）扭曲（调节点可以任意移动位置）：

按住 Ctrl 键调整变换控件框的调节点，可以对图像进行扭曲变形。

（6）透视（调节点的位置对称变化）。

按住 Ctrl＋Shift＋Alt 键调整【变换控件】框的调节点，可以使图像产生透视效果。

2）利用移动工具对图像进行精确变形

选择工具箱中的移动工具，并勾选【显示变换控件】复选框，在图像中显示变换控件框。将鼠标指针移动至变换控件框的调节点或边线上单击，当变换控件框显示为实线框时，选项栏中的内容如图3-5所示。



图 3-5　变换控件选项栏

该选项栏中各选项的功能如下。

- 【参考点位置】：此图标中的黑点表示当前图像中参考点的位置。
- 【设置参考点的水平和垂直位置】：修改其 X、Y 的数值可以精确地定位调节中心的坐标，其单位是像素。
- 【设置水平和垂直缩放比例】：修改 W、H 的数值可以精确地对图像在水平和垂直方向上进行缩放。
- 【保持长宽比】：激活该按钮，锁定水平缩放和垂直缩放使用相同的缩放比例，即使图像为等比缩放。
- 【旋转】：在此文本框中输入数值控制图像旋转的角度。
- 【设置水平和垂直斜切】：修改 H、V 的数值可以控制图像在水平和垂直方向上倾斜的角度。
- 【在自由变换和变形模式之间切换】：激活此按钮，工具选项栏将变成图 3-6 所示的状态。



图 3-6 变形选项栏

【变形】：默认为【自定】选项，此时变换控件将显示为自由变换控件，拖曳自由变换控件的节点、子柄与连线可以对图像进行自由变换。

【取消变换】：单击该按钮，取消对图像的变形操作，也可以按 Esc 键取消。

【进行变换】：单击该按钮，确认对图像进行变形操作。

3. 利用移动工具对图层进行对齐或分布

在移动工具选项栏中利用对齐与分布按钮组也可以对齐和分布图层。

1）对齐图层

选择多个图层，或者选择链接图层中的某个图层之后，可以利用对齐按钮组对齐图层，6 个对齐图层按钮的功能与菜单栏中的【图层】|【对齐】命令中的子菜单命令的功能相同，分别为【顶对齐】、【垂直居中对齐】、【底对齐】、【左对齐】、【水平居中对齐】和【右对齐】。

2）分布图层

选择两个以上的图层或选择链接图层中的某个图层之后，可以利用分布按钮组分布图层，6 个平均分布图层按钮的功能与菜单栏中的【图层】|【分布】命令中的子菜单命令功能相同，分别为【按顶分布】、【垂直居中分布】、【按底分布】、【按左分布】、【水平居中分布】和【按右分布】。

3.1.3 套索工具

套索工具组中包括套索工具、多边形套索工具、磁性套索工具 3 个工具按钮。

1. 套索工具的使用方法和主要功能

（1）套索工具：选择套索工具后，只要按住鼠标左键在图像上拖动，鼠标指针移动的轨

迹就是选区的边界。套索工具的优点是操作简便，缺点是所创建选区的形状较难控制，所以该工具一般用于对精确度要求不高的选择。

(2) 多边形套索工具：选择多边形套索工具后，沿着要选择的图像边界多次单击，新的鼠标指针落点与前一个落点间会出现一条连线，然后将鼠标指针移回起点，当鼠标指针带圈时单击可闭合连线，构成选区。多边形套索工具的优点是选择较精确，缺点是操作时较烦琐，该工具比较适用于边界多为直线或者边界曲折复杂的图案。

(3) 磁性套索工具：它是一种比较特别的选择工具，根据要选择图像边界的像素点颜色来决定选择时的工作方式。在要选择图像边界与背景颜色差别较大的部分可以直接沿边界拖曳鼠标指针，磁性套索工具会根据颜色的差别自动勾画出选框，在颜色差别不大的部分可以用多次单击的方法勾选边界，这一工具主要适用于选择边界分明的图案。

2. 套索工具选项栏

在选择套索工具、多边形套索工具时，选项栏的内容与选框工具选项栏的相似，如图 3-7 所示，比选择磁性套索工具时多了一些参数。

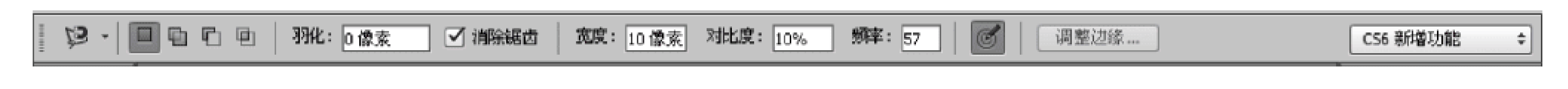


图 3-7　套索工具选项栏

- 【宽度】：该值决定磁性套索工具在多宽的范围内探测图像的边界。磁性套索工具只探测从鼠标指针开始指定距离以内的边缘。
- 【对比度】：该值指定套索对图像中边缘的灵敏度，较高的值将探测与周围对比强烈的边缘，较低的值将探测低对比度的边缘。
- 【频率】：该值决定套索以什么速率设置紧固点，使用较高的数值则捕捉紧固点的速度更快，并且紧固点的数量更多。
- 【使用绘图板压力以更改钢笔宽度】：该选项只有在用户使用绘图板时才起作用。绘图板是一种外部设备，可用手绘的方式向计算机中输入图像。激活该选项后，可以在使用绘图板时以落笔的力度来影响笔画的粗细。

在边缘较明显的图像上可以使用较大的宽度值和较高的对比度值，在边缘较柔和的图像上可以使用较小的宽度值和较低的对比度值。

使用较小的宽度值、较高的对比度值可以进行较精确的选择，使用较大的宽度值、较小的对比度值可进行粗略的选择。

3.1.4　魔棒工具

在工具箱中选择魔棒工具，设置适当的选项后，在要选择的图像上单击，即可选择与鼠标指针落点颜色相近的区域。该工具主要适用于有大块单色区域的图像的选择。

1. 魔棒工具选项栏

在工具箱中选择魔棒工具，选项栏如图 3-8 所示，这里只介绍前面未讲解的选项。

图 3-8　魔棒工具选项栏

- 【容差】：容差取值范围为 0～255，这个参数的值决定了选择的精度，此值越大，选择的精度越低；此值越小，选择的精度越高。
- 【连续】：勾选【连续】复选框，则只能选择与鼠标指针落点处像素颜色相近且相连的部分；取消勾选该复选框，则可以在图像中选择所有与鼠标指针落点处像素颜色相近的部分。
- 【对所有图层取样】：对于多图层的文件来讲，一般情况下所做的操作只对当前图层起作用。在使用魔棒工具时，若不勾选【对所有图层取样】复选框，则只选择当前图层中颜色相近的部分；若勾选该复选框，则可以选择所有图层中可见部分的颜色相近的部分。

2. 快速选择工具

快速选择工具的功能类似于笔刷，并且能够调整圆形笔尖大小绘制选区，在图像中单击并拖动鼠标即可绘制选区，这是一种基于色彩差别但却用画笔智能查找主体边缘的新颖方法。快速选择工具选项栏如图 3-9 所示。



图 3-9　快速选择工具选项栏

- 【选区方式】：3 个按钮从左到右分别是新选区、添加选区、减去选区。当没有选区时，默认的选择方式是新建；在选区建立后，自动改为添加到选区；如果按住 Alt 键，选择方式变为从选区减去。
- 【画笔】：当初选离边缘较远的较大区域时，画笔尺寸可以大一些，以提高选取的效率；但对于小块的主体或修正边缘时则要换成小尺寸的画笔。总体来说，大画笔选择快，但选择粗糙，容易多选；小画笔一次只能选择一小块主体，选择慢，但得到的边缘精度高。更改画笔大小的方法是在建立选区后按]键增大画笔的大小，按[键减小画笔的大小。
- 【自动增强】：勾选此复选框后，可减少选区边界的粗糙度和块效应，即“自动增强”使选区向主体边缘进一步流动并做一些边缘调整。一般勾选此复选框。
- 【对所有图层取样】：当图像中含有多个图层时，勾选该复选框，将对所有可见图层的图像起作用，当没有勾选时，魔棒工具只对当前图层起作用。

3.2　绘画与修饰工具

绘画与修饰工具是用来对位图进行绘制及修改、修饰的工具，包括画笔、渐变、历史记录画笔、修复工具等，它们都是对位图有效、对矢量图无效。

3.2.1 画笔工具、铅笔工具、颜色替换工具和混合器画笔工具

如果用户有一定的手绘功底，可以直接使用画笔工具和铅笔工具绘制图形，使用这两个工具可以创建出不同的效果。另外，用户还可以使用颜色替换工具对照片局部的颜色进行替换。

1. 画笔工具

使用画笔工具可以绘制出边缘柔软的画笔效果，画笔的颜色为工具箱中的前景色。画笔工具选项栏如图 3-10 所示。



图 3-10　画笔工具选项栏

1）画笔选项

（1）单击选项栏中的【画笔】按钮，在列表中可以选择要使用的画笔笔尖。

（2）修改【主直径】值可以设置笔尖的大小。

（3）修改【硬度】值可以修改笔尖边缘的柔化程度。

2）其他选项

（1）在【模式】下拉列表中可以选择画笔工具的模式，不同的模式决定画笔工具使用的颜色以何种方式与图像中的像素进行混合。

（2）【不透明度】值决定画笔的透明度。

（3）【流量】值用来设置颜色随工具移动应用的速度，即设置所绘制线条颜色的流畅程度，它也可以产生一定的透明效果。

在工具箱中选择画笔工具，按住 Shift 键不放，在图像中拖曳鼠标指针，可以创建水平或垂直的线条。

在工具箱中选择画笔工具，在图像中的某一点单击，按住 Shift 键不放，然后在另一点单击，可以在两点间创建一条直线。

2.【画笔】面板

【画笔】面板默认位于面板区内，它提供了大量预置的画笔笔尖形状，并且可以通过设置不同的参数以及选项生成更多的笔尖形状，从而大大增强了 Photoshop 的绘画功能。选择【窗口】|【画笔】命令，或单击面板区左侧的【画笔】按钮，弹出【画笔】面板，如图 3-11 所示。

1）选择预设画笔

用户可以通过两种途径选择当前使用的画笔预设，一种是通过选项栏左侧的画笔弹出式面板进行选择，另一种是通过【画笔】面板进行选择。

2）自定义画笔

在打开的【画笔】面板中选择左侧的【画笔笔尖形状】选项，可以弹出图 3-12 所示的画笔笔尖形状图案。

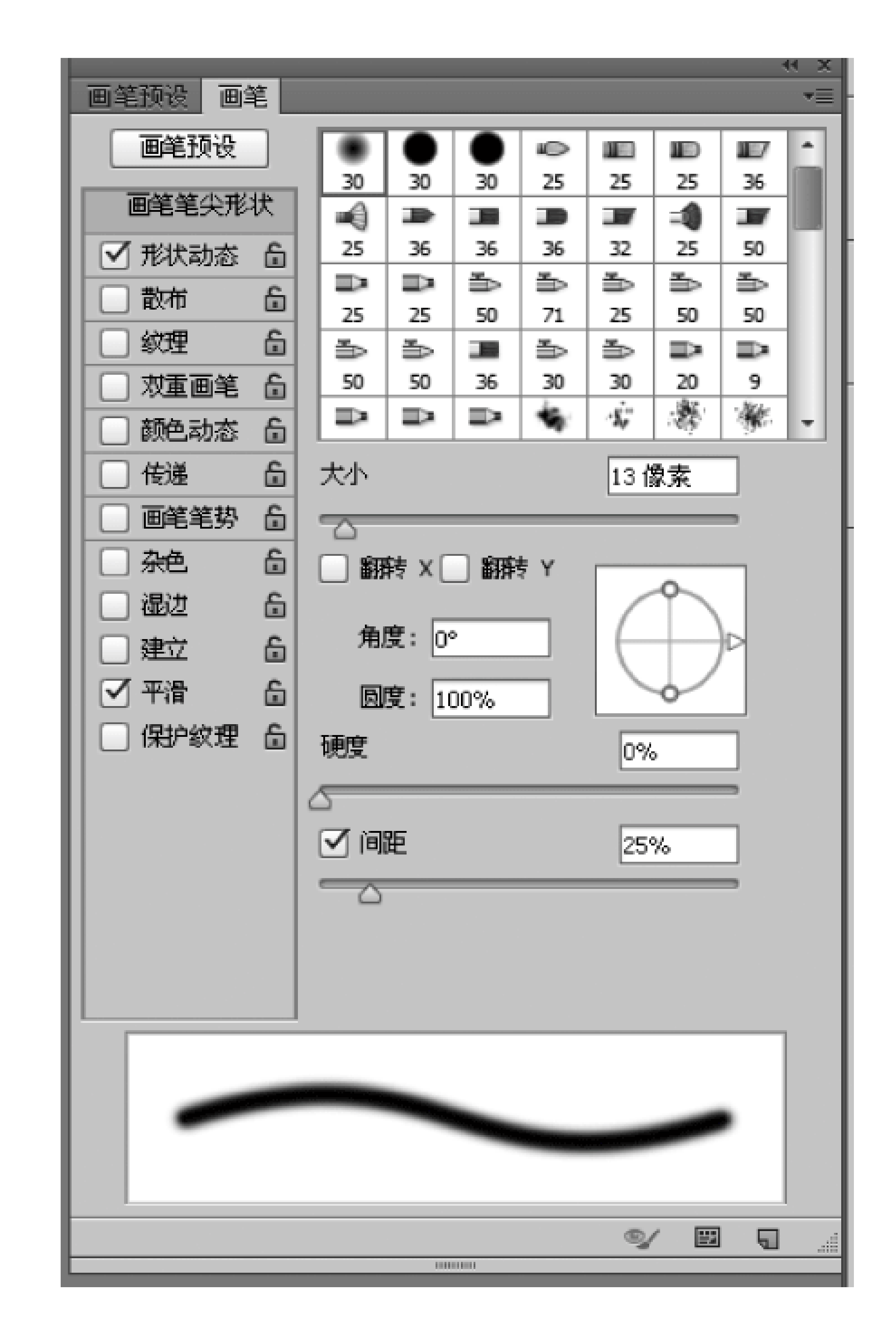


图 3-11　【画笔】面板

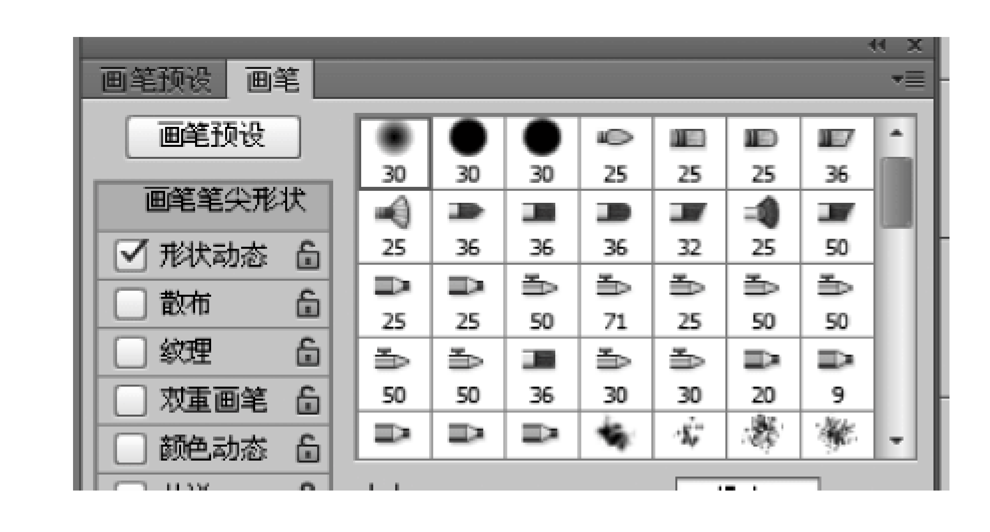


图 3-12　画笔笔尖形状图案

单击【画笔】面板左侧不同的选项名称，在右侧会显示相应的参数面板，通过设置各个不同的参数选项可以创建自定义的画笔笔尖形状。

对于已经预存在【画笔】面板中的各个画笔，可以对其选项进行重新调整，并将调整后的结果利用【新建画笔预设】命令存储为新的画笔。

创建任意规则或不规则的选区，若选区的【羽化】值为 0px，得到的是硬边画笔；如果用户在定义选区的时候设置其他不同的【羽化】值，则得到软边画笔。

自定义画笔形状的大小可高达 2500×2500 像素。为了使画笔的效果更好，最好为画笔设定一个纯白色的背景，这样在用该画笔绘制图形的时候，白色的部分是透明的。在自制画笔时最好使用灰度色彩，画笔颜色是由当前使用的前景色来确定的，这里只定义了画笔的笔尖形状。

3）画笔选项设定

（1）【画笔笔尖形状】类参数：

在【画笔】面板左侧选择【画笔笔尖形状】选项，右侧显示出画笔笔尖形状图案选项，如图 3-11 所示，同时在下方还可以预览设置后的效果。各部分的功能介绍如下。

① 在右侧上方的笔尖形状列表中单击相应的笔尖形状即可将其选择。

② 【直径】值用来控制画笔的大小，用户可以直接修改文本框中的数值，也可以拖动其下方的滑块来改变数值。

③ 在修改【直径】值后，单击【取样大小】按钮，可以将其恢复为默认值（使用圆形笔尖时不显示该按钮）。

④ 勾选【翻转 X】和【翻转 Y】复选框，可以分别将笔尖形状进行水平和垂直翻转。

⑤ 设置【角度】值，则将笔尖以相应的角度逆时针旋转。

⑥ 设置【圆度】值，则将笔尖以相应的比例在竖轴缩放。

⑦【硬度】值只对那些边缘有虚化效果的笔尖有效，【硬度】值越大，画笔边缘越清晰；【硬度】值越小，画笔边缘越模糊、柔和。

⑧ 当勾选【间距】复选框时，其右侧文本框中的值表示画笔每两笔之间跨越画笔直径的百分数，取消勾选时，在图像中画线的形态与拖曳鼠标指针的速度有关，拖曳越快，画笔每两笔之间的跨度就越大，拖曳越慢，画笔每两笔之间的跨度就越小。

(2)【形状动态】类参数：

对于【形状动态】选项下的几个参数，可以在上一步绘制画笔的基础上更加详细地设置画笔的外观，例如【大小抖动】、【控制】、【最小直径】、【角度抖动】、【圆度抖动】、【最小圆度】等，通过对这些参数进行设置可以产生不同的画笔效果。

(3)【散布】类参数：

通过调整【散布】类参数可以设置笔尖沿鼠标指针拖曳的路线向外扩散的范围，从而使画笔工具产生笔触散射效果。在【画笔】面板左侧选择【散布】选项，并将其他选项的勾选取消，通过调节相应参数值的大小可以得到不同的画笔效果。这个选项只适合绘制特殊图形，像星星之类的效果。

(4)【纹理】类参数：

通过设置【纹理】类参数可以在画笔中产生图案纹理效果。

(5)【双重画笔】类参数：

设置【双重画笔】类参数和选项是在已经选好的画笔上再增加一个不同样式的画笔，可以产生两种不同纹理相交的笔尖效果。

(6)【颜色动态】类参数：

设置【颜色动态】类参数可以使笔尖产生两种颜色或图案进行不同程度混合的效果，并且可以调整其混合颜色的色调、饱和度及明亮度等。这类参数的设置在【画笔】面板中看不出笔尖的变化，只有在绘制图像时才能看出效果。

(7)【其他动态】类参数：

【其他动态】类参数可以设置画笔绘制出的颜色的不透明度和使颜色之间产生不同的流动效果。

(8) 其他选项设置：

除了前面介绍的参数和选项外，在【画笔】面板左侧还有以下几个选项。

- 勾选【杂色】复选框可以使画笔产生一些小碎点的效果。
- 勾选【湿边】复选框可以使画笔绘制出的颜色产生中间淡、四周深的润湿效果，可以用来模拟加水较多的颜料产生的效果。
- 勾选【喷枪】复选框可以模拟传统的喷枪，使画笔产生渐变色调的效果。
- 勾选【平滑】复选框可以使画笔绘制出的颜色边缘较平滑。
- 勾选【保护纹理】复选框，当使用【复位画笔】等命令对画笔进行调整时，保护当前画笔的纹理图案不改变。

3. 铅笔工具

使用铅笔工具可以绘制出硬边的线条，如果是斜线，会带有明显的锯齿，绘制的线条颜色为工具箱中的前景色，在工具箱中选择铅笔工具，其选项栏如图 3-13 所示。

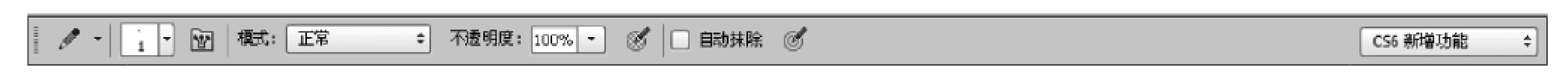


图 3-13　铅笔工具选项栏

铅笔工具的选项设置与画笔工具的基本相同，只是在使用铅笔工具时，【画笔】面板中的所有画笔都不产生虚边放果，所以在【画笔】选项列表中选择【硬度】选项对笔尖效果不起作用。铅笔工具的选项栏比画笔工具多了一个【自动抹除】复选框，如果勾选了【自动抹除】复选框，那么当从图像中使用前景色的像素处落笔时绘出的颜色将为背景色，如果从使用其他颜色的像素处开始落笔，则依然使用前景色。

4. 颜色替换工具

使用颜色替换工具能够简化图像中特定颜色的替换，可以用校正颜色在目标颜色上绘画。在处理照片时，该工具常用来校正图像中较小图像的偏色。颜色替换工具不能在颜色模式为“位图”、“索引”或“多通道”的图像中使用。颜色替换工具选项栏如图 3-14 所示。



图 3-14　颜色替换工具选项栏

5. 混合器画笔工具

使用混合器画笔工具可以绘制出逼真的手绘效果，它是较为专业的绘画工具，通过选项栏的设置可以调节笔触的颜色、潮湿度、混合颜色等，如同用户在绘制水彩或油画的时候随意地调节颜料颜色、浓度、颜色混合等，可以绘制出更为细腻的效果图。其选项栏如图 3-15 所示。

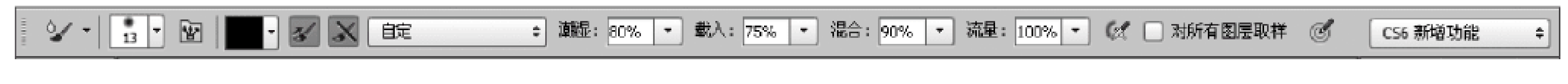


图 3-15　混合器画笔工具选项栏

- 【画笔】：单击该按钮，在打开的下拉列表中调整画笔直径大小以及画笔大小。
- 【显示前景色颜色】：单击右侧三角可以载入画笔、清理画笔、只载入纯色。
- 【每次描边后载入画笔】：每次描边后载入画笔。
- 【每次描边后清理 Photoshop CS6 画笔】：每次描边后清理画笔。

【每次描边后载入画笔】和【每次描边后清理画笔】两个按钮控制了每一笔涂抹结束后对画笔是否更新和清理，类似于画家在绘画时一笔过后是否将画笔在水中清洗。

- 【混合画笔组合】：提供多种为用户提前设定的画笔组合类型，包括干燥、湿润、潮湿和非常潮湿等。【有用的混合画笔组合】下拉列表中是为用户预先设置好的混合画笔。当用户选择某一种混合画笔时，右边的 4 个选择数值会自动改变为预设值。
- 【潮湿】：设置从画布拾取的油彩量，就像是给颜料加水，设置的值越大，画在画布上的色彩越淡。

- 【载入】：设置画笔上的油彩量。
- 【混合】：用于设置多种颜色的混合，当【潮湿】为 0 时，该选项不能用。
- 【流量】：设置描边的流动数。
- 【启用喷枪模式】：作用是当画笔在一个固定的位置一直描绘时，画笔会像喷枪那样一直喷出颜色。如果不启用这个模式，则画笔只描绘一下就停止流出颜色。
- 【对所有图层取样】：作用是无论文件有多少个图层，都将它们作为一个单独的合并图层来看待。
- 【绘图板压力控制大小】：当用户选择普通画笔时，它可以被选择，此时用户可以用绘图板来控制画笔的压力。

3.2.2 渐变工具和油漆桶工具

1. 渐变工具

渐变工具是使用较多的一种工具，使用该工具可以在图像中填充渐变颜色和透明度过渡变化的效果。渐变工具常用来制作图像背景、立体效果和光亮效果等。

在工具箱中选择渐变工具后，其选项栏如图 3-16 所示。

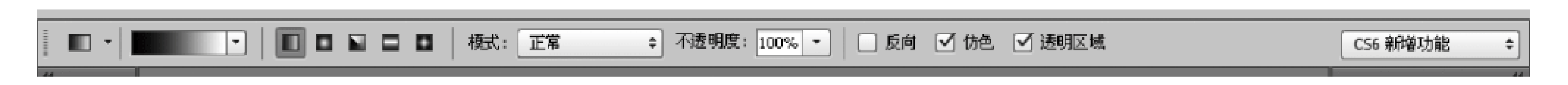


图 3-16 渐变工具选项栏

1）可编辑渐变条

渐变颜色条中显示了当前的渐变颜色，单击其右侧的扩展按钮，可以打开一个弹出式面板，如图 3-17 所示，用户可以在该面板中选择预设的渐变。

单击【可编辑渐变条】，会弹出【渐变编辑器】对话框，如图 3-18 所示。

图 3-17 渐变颜色面板



图 3-18 【渐变编辑器】对话框

- 【名称】：在已有的渐变样式中选择一种渐变作为编辑的基础，在渐变效果预设条中调节任何一个项目后，“名称”自动变成“自定”，用户可以自定名称。
- 【渐变类型】：有实底、杂色两种，如为实底，则能对平滑度、色标的位置、颜色、不透明度等方式进行调整；如为杂色，则可以设置渐变的粗糙度和颜色模式。

2）渐变类型

渐变类型包括线性渐变、径向渐变、角度渐变、对称渐变和菱形渐变 5 种。使用时单击所需渐变类型对应的按钮，然后在图像中绘制即可。

- 【线性渐变】：从起点到终点以直线渐变。
- 【径向渐变】：从起点到终点以圆形图案渐变。
- 【角度渐变】：围绕起点以逆时针方向环绕渐变。
- 【对称渐变】：在起点两侧产生对称直线渐变。
- 【菱形渐变】：从起点到终点以菱形图案渐变。

3）模式

【模式】选项用来设置应用渐变时的混合模式。

4）不透明度

【不透明度】选项用来设置渐变效果的不透明度。

5）反向

勾选【反向】复选框可转换渐变条中的颜色顺序，得到反向的渐变效果。

6）仿色

【仿色】复选框用来控制色彩的显示，勾选它可以使色彩的过渡更平滑。

7）透明区域

勾选该复选框，可创建透明渐变；取消勾选，则只能创建 Photoshop CS6 实色渐变。

2. 油漆桶工具

在工具箱中专门提供了一个填充颜色的工具，即油漆桶工具，它按照图像中像素的颜色进行填充色处理，其填充范围是与鼠标指针落点所在像素点的颜色相同或相近的像素点。在工具箱中选择油漆桶工具，其选项栏如图 3-19 所示。



图 3-19　油漆桶工具选项栏

在工具箱中选择油漆桶工具，然后在图像中单击即可填充前景色或图案。

3.2.3　历史记录画笔工具和历史记录艺术画笔工具

使用历史记录画笔工具和历史记录艺术画笔工具可以在图像中将新绘制的部分恢复到【历史记录】面板中的“恢复点”处的状态。

1. 历史记录画笔工具

历史记录画笔工具的功能有点类似【历史记录】面板，也可以撤销前面的操作，其选项栏

如图 3-20 所示。



图 3-20　历史记录画笔工具选项栏

在【历史记录】面板中设置好恢复点的位置，使用工具在图像中拖曳，即可将鼠标拖过的部分恢复到恢复点的状态。使用历史记录画笔工具恢复图像优于【历史记录】面板，可以有选择地擦除多余的操作。用户可以在局部拖曳鼠标指针进行恢复，或在建立的选区内进行恢复，也可以通过修改【历史记录】面板中恢复点的位置将一幅图像的不同部分恢复到不同的状态。

2. 历史记录艺术画笔工具

历史记录艺术画笔工具的使用方法与历史记录画笔工具基本相同，只是使用历史记录艺术画笔工具恢复图像时，在将图像恢复到恢复点处效果的同时对图像像素进行了移动和涂抹，使图像产生一种被涂花的效果，其选项栏如图 3-21 所示。



图 3-21　历史记录艺术画笔工具选项栏

- 【样式】：该下拉列表中包含 10 种移动和涂抹图像像素的方式。
- 【区域】：在该文本框中设置在多少像素范围内进行移动和涂抹。
- 【容差】：在该文本框中设置当前图像与恢复点图像颜色之间有多大的差异，以便进行移动和涂抹。当【容差】值为 0 时，可在图像中的任何地方进行移动和涂抹；当【容差】值较大时，只能在与恢复点颜色明显不同的区域进行移动和涂抹。

3.2.4　修复工具组

Photoshop 加强了照片处理功能。在工具箱中专门用于修复旧照片的工具共有 5 个，即污点修复画笔工具、修复画笔工具、修补工具、红眼工具、内容感知移动工具。污点修复画笔工具和修补工具主要用于在保持原图像明暗效果不变的情况下消除图像中的杂色、斑点，红眼工具主要用于处理照片中出现的红眼问题。

1. 污点修复画笔工具

使用污点修复画笔工具可以快速移去照片中的污点和其他不理想部分。污点修复画笔工具将自动从所修饰区域的周围取样，使用取样点周围图像或图案中的样本像素进行绘画，并将样本像素的纹理、光照、透明度和阴影与所修复的像素相匹配。污点修复画笔工具选项栏如图 3-22 所示，该工具常用于对图像中面积相对较小的污点进行修复，如果修饰大片区域或需要更大程度地控制来源取样，则使用修复画笔工具的效果会更好。



图 3-22　污点修复画笔工具选项栏

2. 修复画笔工具

修复画笔工具和污点修复画笔工具类似，但是修复画笔工具使用指定的图像取样点来修复图像中的缺陷，或复制预先设置好的图案到需要修复的位置，且将复制过来的图像或图案边缘虚化，并与要修复的图像按指定的模式进行混合，混合的图像不改变需要修复图像的明暗，从而达到最佳的修复效果，其选项栏如图 3-23 所示。

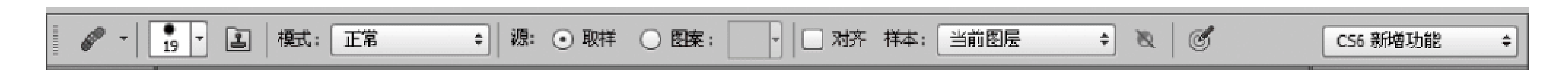


图 3-23　修复画笔工具选项栏

【源】有两个可选项，选择【取样】单选按钮时是利用从图像中定义的图像进行修复，选择【图案】单选按钮时是利用右侧的【图案】面板中的图案对图像进行修复。

在进行污损照片的处理时，污点修复画笔工具主要用于消除图像中小范围内的杂点、划痕或者污渍等，但修复画笔工具不仅能用于污损照片的处理，还可以用于其他方面。

用相同的方式修复图像，可以在修复的过程中多次重新指定取样点。

3. 修补工具

修补工具的功能与修复画笔工具相似，修补工具类似一个增加了选择功能的修复画笔工具。在使用修补工具修补图像时，称需要修补的图像为源图像，把用来修改源图像的图像称为目标图像。

修补工具可以用来选取大面积的图像进行修复，其选项栏如图 3-24 所示。

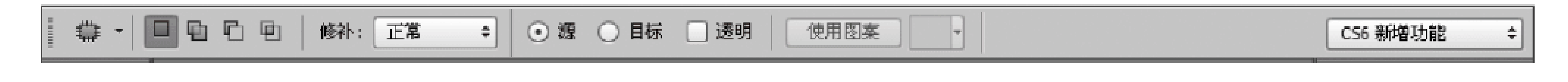


图 3-24　修补工具选项栏

【新选区】按钮、【添加到选区】按钮、【从选区减去】按钮、【与选区交叉】按钮的功能与选框工具选项栏中按钮的功能相同。

- 【源】：选择该单选按钮后，可在图像中选择要修复的部分，然后将其拖曳至相似的图像处进行修复。
- 【目标】：选择该单选按钮后，可在图像中选择与要修复的图像相似的部分，然后将其拖曳至要修复的图像处进行修复。

修补工具和修复画笔工具都是用来修补图像的，修复画笔工具主要用于对细节进行修改，修补工具主要用于对较大范围图像的修改，如果修补的图像有些过渡不自然的地方，可以使用修复画笔做进一步修补。

4. 红眼工具

红眼工具使用前景色对图像中特定的颜色进行替换。在用于照片处理时，该工具常用来校正图像中较小图像的偏色，例如在拍摄夜间的照片时，人或动物的眼睛经常会因为反光出现红色，被称为“红眼”。使用红眼工具可以方便地消除红眼问题，也可以移去用闪光灯拍摄的动物照片中的白色或绿色反光。红眼工具不能在颜色模式为“位图”、“索引”或“多通

道”的图像中使用。

在工具箱中选择红眼工具后，选项栏如图 3-25 所示。



图 3-25　红眼工具选项栏

- 【瞳孔大小】：在此文本框中设置修复后瞳孔（眼睛暗色的中心）与眼珠的比例，数值越大，瞳孔越大。
- 【变暗量】：在此文本框中设置修复后瞳孔的暗度，数值越大，瞳孔越暗。

5. 内容感知移动工具

内容感知移动工具可以简单到只需选择图像场景中的某个物体，然后将其移动到图像中的任何位置，经过 Photoshop 的计算，完成极其真实的 PS 合成效果。

- 【感知移动】：这个功能主要用来移动图片中的主体，并随意放置到合适的位置，对于移动后的空隙位置，PS 会智能修复。
- 【快速复制】：选取想要复制的部分，移到其他需要的位置就可以实现复制，复制后的边缘会自动进行柔化处理，以跟周围环境相融合。

在工具箱的修复工具组中选择内容感知移动工具，用户的鼠标指针上就会出现“X”图形，按住鼠标左键并拖动就可以画出选区，和套索工具的操作方法一样。用户先用这个工具把需要移动的部分选取出来，然后在选区中按住鼠标左键拖动，移到想要放置的位置后松开鼠标系统就会智能修复。其选项栏如图 3-26 所示。



图 3-26　内容感知移动工具选项栏

3.2.5　图章工具组

图章工具组中包括仿制图章工具和图案图章工具，它们主要是通过在图像中选择印制点或设置图案对图像进行复制。

1. 仿制图章工具

使用仿制图章工具可以准确地复制图像的一部分或全部。仿制图章工具的操作与修复画笔工具相似，按住 Alt 键不放，在图像中要复制的部分单击，即可取得这部分作为样本，在目标位置处单击或拖曳鼠标指针，即可将取得的样本复制到目标位置。其选项栏如图 3-27 所示。

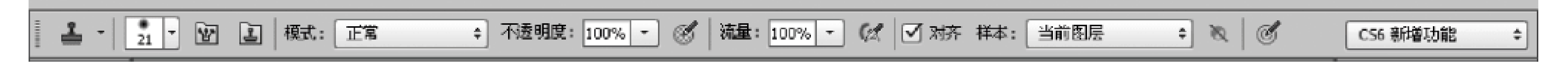


图 3-27　仿制图章工具选项栏

在进行不对齐复制时，如果想再复制其他部分的图像，只要按住 Alt 键，在需要复制的图像上重新定义一个起点就可以了。使用仿制图章工具复制图像可以在一幅图像中进行，

也可以在多幅图像之间进行。

2．图案图章工具

使用图案图章工具不是复制图像中的内容，而是复制已有的图案，其选项栏如图3-28所示。



图3-28　图案图章工具选项栏

单击【图案】按钮，或单击【图案】按钮右上角的按钮，可以利用弹出的下拉菜单中的命令设置【图案】面板，设置【图案】面板的命令与设置【画笔】面板的命令相似。

- 【对齐】：勾选该复选框，在图像窗口中多次拖曳鼠标指针，复制的图案整齐排列；若不勾选该复选框，在图像窗口中多次拖曳鼠标指针，复制的图案将无序地散落在图像窗口中。
- 【印象派效果】：勾选该复选框，复制的图案会产生扭曲模糊的效果。

使用图案图章工具可以将选定的图案复制到一幅或多幅图像中，并且在复制的过程中可以随时通过选项栏在【图案】面板中选择其他图案。

3．自定义图案

Photoshop中提供的图案并不多，为了方便用户使用，Photoshop还专门提供了让用户自定义图案的功能。

3.2.6　橡皮擦工具组

Photoshop工具箱中提供了橡皮擦工具、背景橡皮擦工具、魔术橡皮擦工具，它们的主要功能是在图像中清除不需要的图像像素，以对图像进行修整。

1．橡皮擦工具

橡皮擦工具是基本的擦除工具，它的功能就像橡皮，在使用橡皮擦工具时，如果当前层是背景层，那么被擦除的图像位置显示背景色，这时可以把橡皮擦工具看成是使用背景色作画的绘画工具；如果当前层是普通图层，被擦除的图像位置显示透明放果。在工具箱中选择橡皮擦工具，其选项栏如图3-29所示。

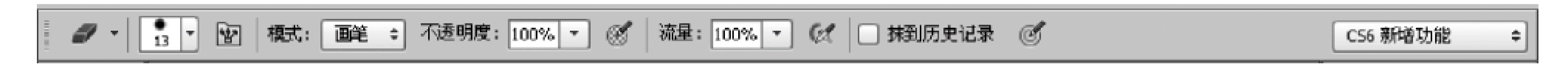


图3-29　橡皮擦工具选项栏

【模式】下拉列表中共有3个选项，橡皮擦工具选项栏中的其他内容会随【模式】选项的不同而产生相应的变化。

当选择【画笔】和【铅笔】选项时，橡皮擦工具的选项和使用方法与画笔工具、铅笔工具相似，只不过在背景层上使用时所用的颜色为背景色，在普通层上使用时产生的效果为透明。

当选择【块】选项时，橡皮擦工具在图像窗口中的大小是固定不变的，所以可以先将图像

放大到一定倍数，然后再利用它对图像中的细微处进行修改。当图像被放大到3200%时，橡皮擦工具的大小恰好是一个像素的大小，这时可以对图像进行精确到一个像素的修改。

若勾选【抹到历史记录】复选框，使用橡皮擦工具可将图像擦除至【历史记录】面板中恢复点处的图像效果，这有点类似于历史记录画笔工具的功能。

2. 背景橡皮擦工具

使用背景橡皮擦工具可以将图像中特定的颜色擦除。在擦除时，如果当前层是背景层，Photoshop会自动将其转换为普通层，也就是说，使用背景橡皮擦工具可以将图像擦除至透明。

在工具箱中选择背景橡皮擦工具，其选项栏如图3-30所示。

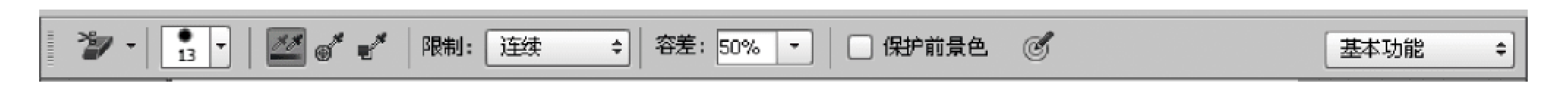


图3-30 背景橡皮擦工具选项栏

【连续】：激活该按钮，背景橡皮擦工具擦除笔尖中心经过的像素颜色，当笔尖中心经过某一像素时，该像素的颜色被指定为背景色。

【一次】：激活该按钮，背景橡皮擦工具擦除鼠标指针落点处像素的颜色，该落点处像素的颜色被设置为背景色，只要一直拖曳鼠标指针将一直擦除这一颜色。

【背景色板】：激活该按钮，可以在工具箱中先将背景色设置为需要擦除的颜色，然后在图像中拖曳鼠标指针，只擦除指定的背景色。

【限制】：该下拉列表中的选项决定了背景橡皮擦工具的作用范围。

- 选择【不连续】选项，擦除笔尖拖过的范围内所有与指定颜色相近的像素。
- 选择【连续】选项，擦除笔尖拖过的范围内所有与指定颜色相近且相连的像素。
- 选择【查找边缘】选项与选择【连续】选项的功能相似，只是选择【查找边缘】选项会在图像中保留较强的边缘效果。

【容差】：决定在图像中选择要擦除颜色的精度，此值越大，可擦除颜色的范围就越大；此值越小，可擦除颜色的范围就越小。

【保护前景色】：勾选该复选框，图像中使用前景色的像素不被擦除。

3. 魔术橡皮擦工具

魔术橡皮擦工具与前面介绍的两种橡皮擦工具在操作上有所不同，前面的两种橡皮擦工具通常是在图像中拖曳鼠标指针，而使用魔术橡皮擦工具，只要在图像中需要擦除的颜色上单击，即可在图像中擦除与鼠标指针落点处颜色相近的像素。魔术橡皮擦工具的选项、使用方法和功能有些类似魔术棒工具，使用魔术橡皮擦只擦除图像中颜色相近的像素，而使用魔术棒工具选择图像中颜色相近的像素。在工具箱中选择魔术橡皮擦工具，其选项栏如图3-31所示。



图3-31 魔术橡皮擦工具选项栏

3.2.7　模糊工具、锐化工具和涂抹工具

1. 模糊工具

模糊工具主要用于对图像进行柔化模糊。

在工具箱中选择模糊工具，其选项栏如图 3-32 所示。

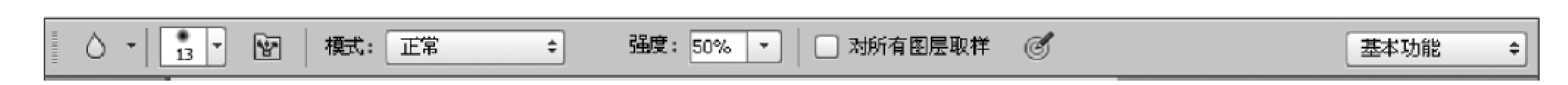


图 3-32　模糊工具选项栏

选项栏上的【强度】值决定了每当拖曳鼠标指针时可以使图像达到的模糊程度。

2. 锐化工具

锐化工具主要用于对图像进行锐化。

3. 涂抹工具

在图像中拖曳鼠标指针，可以将鼠标指针落点处的颜色抹开，其作用是模拟刚画好一幅画还没干时用手指去抹的效果。

3.2.8　减淡工具、加深工具和海绵工具

1. 减淡工具

减淡工具主要用于对图像的阴影、半色调及高光等部分进行提亮、加光处理。

在工具箱中选择减淡工具，其选项栏如图 3-33 所示。



图 3-33　减淡工具选项栏

在选项栏的【范围】下拉列表中有 3 个选项。

- 选择【阴影】选项，对图像中较暗的部分起作用。
- 选择【中间调】选项，平均地对整个图像起作用。
- 选择【高光】选项，对图像中较亮的部分起作用。

减淡工具选项的【曝光度】值决定了一次操作对图像的提亮程度。

2. 加深工具

加深工具主要用于对图像的阴影、半色调及高光等部分进行遮光、变暗处理。

加深工具的选项栏与减淡工具的基本相同，只是它的【曝光度】值决定了一次操作对图像的遮光程度。

3. 海绵工具

海绵工具主要用于对图像进行变灰或提纯（也就是使图像颜色更加鲜艳）处理。

在工具箱中选择海绵工具，其选项栏如图 3-34 所示。

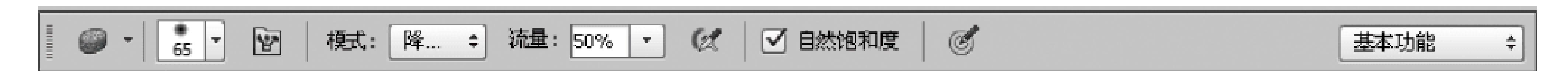


图 3-34　海绵工具选项栏

在海绵工具选项栏的【模式】下拉列表中有两个选项。

- 选择【降低饱和度】选项，海绵工具降低图像色彩的饱和度，对图像进行变灰处理。
- 选择【饱和】选项，海绵工具提高图像色彩的饱和度，对图像进行提纯处理。

3.3　路径和矢量工具

路径和矢量工具是用来制作矢量图的工具，钢笔和矢量形状工具用来绘制矢量形状，其他路径选择工具用来对路径进行编辑修改。

3.3.1　路径的构成

路径是由多个锚点组成的矢量线条，它并不是图像中真实的像素，而是一种绘图的依据。使用 Photoshop 提供的路径创建及编辑工具可以编辑制作出各种形状的路径。路径一般用于对图案的描边、填充以及与选区的转换等，其精确度高，便于调整，人们常使用路径功能来创建一些特殊形状的图像效果。图 3-35 所示为路径构成说明图，其中平滑点和角点都属于路径的锚点。

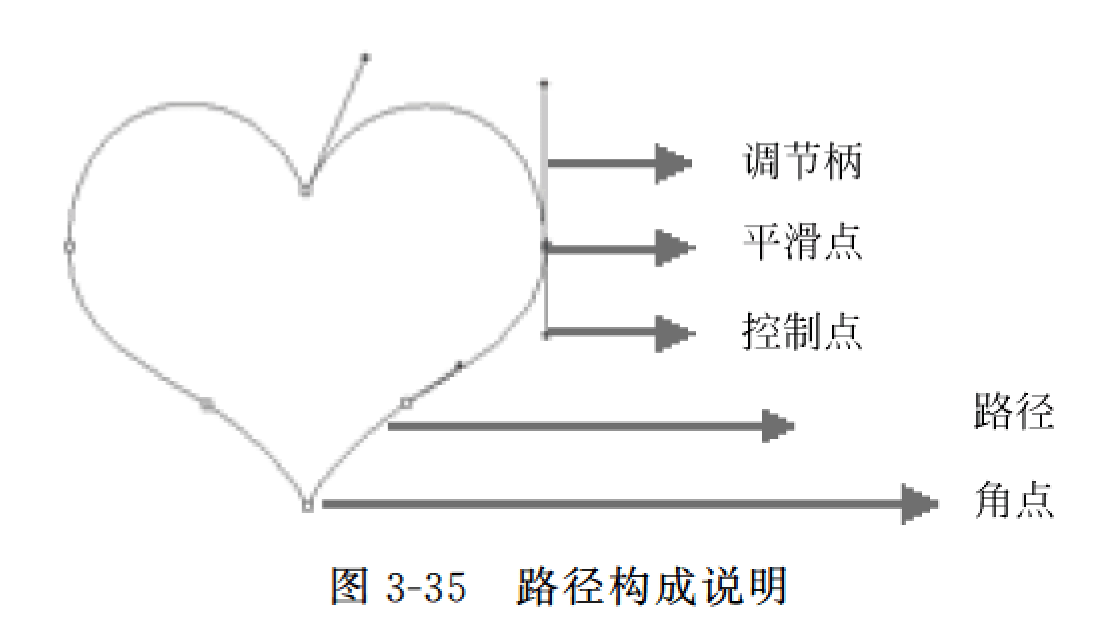


图 3-35　路径构成说明

- 锚点：路径上有一些矩形的小点，人们称之为锚点。锚点标记路径上线段的端点，通过调整锚点的位置和形态可以对路径进行各种变形调整。
- 平滑点和角点：路径中的锚点有两种，一种是平滑点，另一种是角点，平滑点两侧的调节柄在一条直线上，而角点两侧的调节柄不在一条直线上。直线组成的路径没有调节柄，但也属于角点。
- 调节柄和控制点：当平滑点被选择时，其两侧各有一条调节柄，调节柄两边的端点为控制点，移动控制点的位置可以调整平滑点两侧曲线的形态。
- 工作路径和子路径：路径的全称是工作路径，一个工作路径可以由一个或多个子路径构成。在图像中每一次使用钢笔工具或自由钢笔工具创建的路径都是一个子路径。在完成所有子路径后，可以使用选项栏中的选项将创建的子路径组成新的工作路径。Photoshop 提供的创建、编辑路径的工具有两组，一组是钢笔工具，包括钢笔

工具、自由钢笔工具、添加锚点工具、删除锚点工具和转换点工具，这组工具主要用于对路径进行创建和编辑；另一组是路径选择工具，包括路径选择工具和直接选择工具，这组工具主要用于对路径和路径上的控制点进行选择，并进行编辑。

3.3.2　钢笔工具和自由钢笔工具

1. 钢笔工具

钢笔工具主要用于在图像中创建工作路径或形状。

1）钢笔工具的基本操作

首先在工具箱中选择钢笔工具，创建路径的基本操作有以下几种：

- 选择钢笔工具，在图像中移动鼠标指针到需要的位置连续单击，即可创建由线段构成的路径。
- 在图像中移动鼠标指针到需要的位置单击并拖曳，调整路径形态到需要的状态后释放鼠标左键，即可创建锚点为平滑点的曲线路径。
- 当拖曳出调节柄后，按住 Alt 键进行拖曳，即可创建锚点为角点的曲线路径。
- 在创建了一段路径后，将鼠标指针移动到创建起点，当鼠标指针带句号形状时，单击即可闭合路径。

2）钢笔工具的选项

在工具箱中选择钢笔工具，选项栏如图 3-36 所示。



图 3-36　钢笔工具选项栏

（1）【类型】下拉列表中包括形状、路径、像素，每个选项对应的工具选项不同（选择矩形工具后，【像素】选项才可用）。

- 【形状】选项：在图像中可同时创建形状与路径。
- 【路径】选项：在图像中只创建新的工作路径，并将路径保存到【路径】面板中。
- 【像素】选项：在使用钢笔工具时，该选项处于不可用状态。钢笔工具选项栏是与形状工具共用的，只有在使用形状工具时该选项才可用。

（2）【建立】中有选区、蒙版、形状 3 个按钮，在绘制完路径后，单击【选区】按钮会弹出【建立选区】对话框，在其中设置完参数，单击【确定】按钮，则将路径转换为选区；单击【蒙版】按钮则会生成矢量蒙版；单击【形状】按钮则将绘制的路径转换为形状图层。

（3）【绘制模式】：其用法与选区相同，可以实现路径的相加、相减和相交等运算。

（4）【对齐方式】：可以设置路径的对齐方式（当文件中有两条以上路径被选择的情况下可用），与文字的对齐方式类似。

（5）【排列顺序】：设置路径的排列方式。

（6）【橡皮带】：可以设置路径在绘制的时候是否连续。

（7）【自动添加/删除】：如果勾选该复选框，当将钢笔工具移动到锚点上时，钢笔工具会自动转换为删除锚点样式；当移动到路径线上时，钢笔工具会自动转换为添加锚点的样式。

(8)【对齐边缘】：将矢量形状边缘与像素网格对齐(选择【形状】选项时，对齐边缘可用)。

2. 自由钢笔工具

使用自由钢笔工具可以创建形态随意的不规则曲线路径。它的优点是操作较简便；缺点是不够精确，而且经常会产生过多的锚点。

1) 自由钢笔工具的基本操作

在工具箱中选择自由钢笔工具，在图像中拖曳鼠标指针，沿鼠标指针拖过的轨迹会自动生成路径；如果将鼠标指针移动到起点，当鼠标指针带有句号形状时，单击可以闭合路径。

选择自由钢笔工具，在图像中拖曳鼠标指针，在未闭合路径前按住 Ctrl 键后松开鼠标，可以直接在当前位置到路径起点生成直线线段闭合路径。

2) 自由钢笔工具的选项

自由钢笔工具和钢笔工具的选项栏基本相似，如图 3-37 所示，下面只介绍与钢笔工具不同的部分。

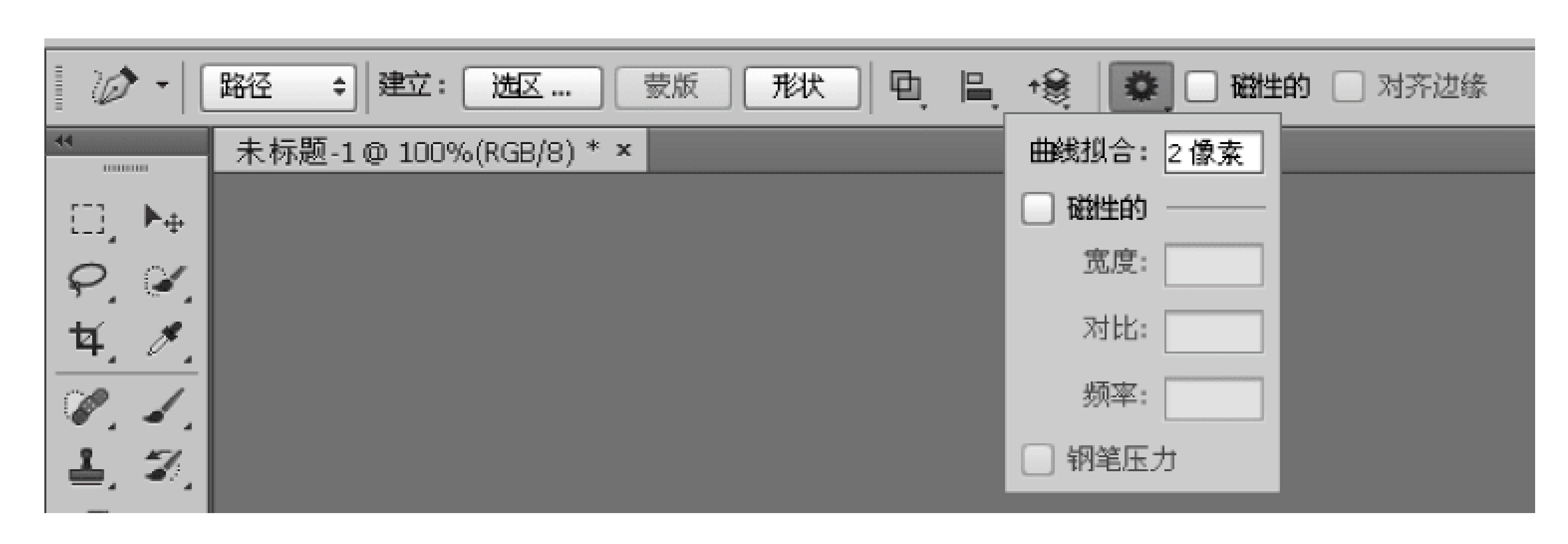


图 3-37 自由钢笔工具选项栏

- 【曲线拟合】：数值越大(10 像素为最大)，锚点越少；数值越小(0.5 像素为最小)，锚点越多。
- 【宽度】：可以调整路径选择范围，数值越大，选择的范围越大。
- 【对比】：对图像边缘的灵敏度，较大的值只能探测与周围强烈对比的边缘，较小的值则探测低对比度的边缘。
- 【频率】：值越大，绘制路径时产生的锚点越多。
- 【钢笔压力】：在使用绘图板输入图像时，根据光笔的压力改变“宽度”值。

3.3.3 添加锚点工具和删除锚点工具

如果没有在钢笔工具选项栏中勾选【自动添加/删除】复选框，那么可以选择工具箱中的添加锚点工具在路径上单击添加锚点，之后选择工具箱中的删除锚点工具在锚点上单击可以将其删除。

3.3.4 转换点工具

路径上的锚点有两种类型，即角点和平滑点，这两者可以相互转换。选择工具箱中的转换点工具，单击路径上的平滑点可以将其转换为角点，拖曳路径上的角点可以将其转换为平

滑点。

3.3.5　路径选择工具和直接选择工具

1. 路径选择工具的选项与功能

使用工具箱中的路径选择工具可以对路径和子路径进行选择、移动、对齐和复制等操作。当子路径上的锚点全部显示为黑色时，表示该子路径被选择。

在选择路径选择工具后，其选项栏如图 3-38 所示。

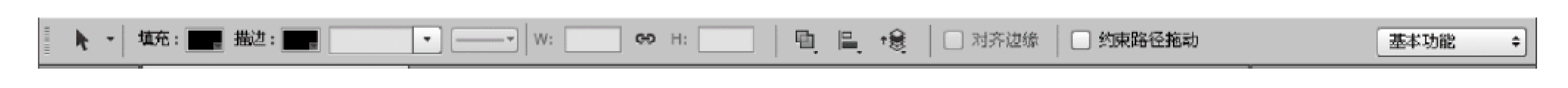


图 3-38　路径选择工具选项栏

- 【填充】、【描边】、【描边宽度】、【描边类型】：这 4 个选项在前面介绍钢笔工具时选择【形状】后可用。
- W、H：路径的高度和宽度，可输入数值更改 Photoshop CS6 路径的宽度和高度。
- 对齐边缘：在绘制路径时可自动对齐网格，选择 Photoshop CS6 菜单栏中的【窗口】|【显示】|【网格】命令打开网格，方便绘制路径时使用。
- 【约束路径拖动】：它只会针对用户所选择的一段路径进行更改，而不会影响其他段路径。

(1) 在图像窗口中拖曳鼠标指针，鼠标指针拖曳范围内的子路径可同时被选择。

(2) 按住 Shift 键，依次单击子路径，可以选择多个子路径。

(3) 在图像窗口中拖曳被选择的子路径，可以进行移动。

(4) 按住 Alt 键拖曳被选择的子路径，可以将被选择的子路径进行复制。

(5) 拖曳被选择的子路径至另一个图像窗口中，可以将子路径复制到另一个图像文件中。

(6) 按住 Ctrl 键，在图像窗口中选择路径，则路径选择工具将被转换为直接选择工具。

2. 直接选择工具的功能

直接选择工具没有选项栏，使用直接选择工具可以选择和移动路径、锚点以及平滑点两侧的控制点。使用直接选择工具对路径和锚点进行的操作有以下几种：

(1) 单击子路径上的锚点可以将其选择，被选择的锚点将显示为黑色。

(2) 在子路径上拖曳鼠标指针，鼠标指针拖曳范围内的锚点可以同时被选择。

(3) 按住 Shift 键依次单击锚点，可以选择多个锚点。

(4) 按住 Alt 键单击子路径，可以选择整个子路径。

(5) 在图像中拖曳两个锚点间的一段路径可以直接调整这一段路径的形态和位置。

(6) 在图像窗口中拖曳被选择的锚点可以移动该锚点的位置。

(7) 拖曳平滑点两侧的控制点可以改变其两侧曲线的形态。

(8) 按住 Ctrl 键在图像窗口中选择路径，则直接选择工具将被转换为路径选择工具。

3.3.6 矢量图形工具

矢量图形工具主要包括矩形工具、圆角矩形、椭圆工具、多边形工具、直线工具和自定形状工具，它们的使用方法非常简单，在工具箱中选择相应的形状工具后，在图像文件中拖曳鼠标指针即可绘制出需要的矢量图形。

1. 形状工具的基本选项

在此以矩形工具为例来介绍形状工具的选项功能。

矩形工具选项栏中的内容与钢笔工具相似，单击选项栏最右侧的按钮，在弹出的面板中可以对样式进行选择，如图 3-39 所示。



图 3-39 形状工具选项栏

2. 形状工具的其他选项

每一种形状工具除了具有它们共同的基本选项以外，还有一些个性选项。

1）矩形工具选项

选择矩形工具，可以在图像中创建矩形效果，此时单击选项栏中右侧的按钮将弹出【矩形选项】面板。

2）圆角矩形工具选项

选择圆角矩形工具，可以在图像中创建圆角矩形效果，在圆角矩形工具选项栏中会多出一个【半径】选项，用于设置圆角半径。

3）椭圆工具选项

选择椭圆工具，可以在图像中创建椭圆效果，此时单击选项栏中右侧的按钮将弹出【椭圆选项】面板。

4）多边形工具选项

选择多边形工具，选项栏中多出一个【边】选项，可以设置要创建多边形的边数，此时单击选项栏右侧的按钮将弹出【多边形选项】面板。

- 勾选【平滑拐角】复选框，创建的多边形的角将产生光滑圆角的效果。
- 勾选【星形】复选框，产生的不是多边形而是多角星形的效果。
- 勾选【星形】复选框后，可以在【缩进边依据】文本框中设置产生多角星形时边的缩进程度，勾选【平滑缩进】复选框，多角星形缩进的角将产生平滑圆角效果。

5）直线工具选项

选择直线工具，可以在图像中创建直线和箭头效果，此时选项栏中有一个【粗细】选项，可以用来设置线宽，单击选项栏右侧的按钮将弹出【箭头】面板。

- 勾选【起点】复选框，在开始画线时添加箭头效果。

- 勾选【终点】复选框，在结束画线时添加箭头效果。
- 【宽度】值为箭头尾端的宽度与线宽的百分比。
- 【长度】值为箭头的长度与线宽的百分比。
- 【凹度】值表示箭头尾端的凹陷程度。当此值为正时，箭头尾端向内凹陷；当此值为负时，箭头尾端向外凸出；当此值为“0”时，箭头尾端平齐。

6）自定形状工具选项

选择自定形状工具，可以在图像中创建自定义的形状效果，单击选项栏右侧的按钮将弹出【自定形状选项】面板。

在选项栏中单击【自定形状】按钮后，选项栏中将多出一个【形状】按钮，单击【形状】按钮，可以在弹出的【形状】面板中选择需要的自定义形状效果。

3.3.7 【路径】面板

前面讲过路径不是图像中的真实像素，而是一种绘图的依据，对路径的描边和填充是通过【路径】面板进行的。

在【路径】面板中除了可以描边和填充路径外，还可以对路径与选区进行转换，对路径进行新建、复制及删除等操作，这些功能大大提高了制作路径的灵活性。由于使用路径制作图像和建立选区的精确度较高且便于调整，因此在图像处理中的应用非常广泛。

1.【路径】面板的基本操作

当前文件中的工作路径堆叠在【路径】面板上方，其中左侧为路径的缩略图，显示路径的缩览效果，右侧为路径的名称。

- 在【路径】面板中直接拖曳路径至相应的位置就可以移动该路径的堆叠位置。
- 在【路径】面板中单击相应的路径就可以将路径打开，使其在图像窗口中显示，以便于进行各种操作。
- 单击【路径】面板堆叠下方的空白处，可以隐藏路径，使其不在图像窗口中显示。
- 双击路径的名称可以对其重新命名，在修改的同时就将该路径保存。

2.【路径】面板菜单

单击【路径】面板右上角的【扩展】按钮，将弹出图 3-40 所示的面板菜单。

1）【新建路径】和【存储路径】命令

如果当前选择的是已保存的路径，那么【路径】面板菜单中显示【新建路径】命令。选择【新建路径】命令，在弹出的对话框的【名称】文本框中设置新路径的名称，此时新创建的路径自动保存。

如果当前选择的是未保存的路径，那么【路径】面板菜单中【新建路径】命令的位置处显示的则是【存储路径】命令。选择【存储路径】命令，在弹出的对话框的【名称】文本框中设置要存储路径的名称，



图 3-40　【路径】面板及面板菜单

单击【确定】按钮将当前工作路径保存。

2）【复制路径】命令

只有当【路径】面板中选择的是已保存的路径时，【复制路径】命令才可用。选择【复制路径】命令，在弹出的对话框的【名称】文本框中设置新复制路径的名称，单击【确定】按钮可以将当前路径复制。

3）【删除路径】命令

选择【删除路径】命令，可以将当前被选择的路径删除。

4）【建立工作路径】命令

只有当图像中存在选区时，【建立工作路径】命令才可用。选择该命令，在弹出的对话框中设置容差的值，单击【确定】按钮。其中【容差】值用来设置将选区转换为路径的精确程度。

5）【建立选区】命令

在图像中选择路径并选择【建立选区】命令，弹出【建立选区】对话框，在对话框中进行设置。

6）【填充路径】命令

选择要进行填充的路径，并选择【填充路径】命令。

7）【描边路径】命令

选择要进行描边的路径，然后选择【描边路径】命令，在弹出的对话框的【工具】下拉列表中选择描边所要使用的工具。

8）【剪贴路径】命令

【剪贴路径】的主要功能是在图像进行打印和输出到 Illustrator 或 InDesign 中时设置图像的透明区域，这一功能在 Photoshop 中不起作用，所以读者只要大概了解就可以了。注意要作为剪贴路径的路径必须是已经保存的路径。

9）【面板选项】命令

选择该命令，可以在弹出的【路径面板选项】对话框中设置缩略图的大小。

3. 创建新路径和创建子路径

在图像中新创建的路径实际上是一个未保存的工作路径，一个工作路径可以由一个子路径组成，也可以由多个子路径组成，那么在使用钢笔工具创建路径时所创建的到底是一个什么样的工作路径呢？

1）当图像中只存在一个未保存的路径时

- 如果该路径处于隐藏状态，那么在图像中使用钢笔工具创建一个新路径并将原有的路径删除。
- 如果该路径不处于隐藏状态，此时在图像中使用钢笔工具创建一个新路径相当于继续编辑当前的工作路径，也就是说相当于创建子路径。

2）当图像中存在一个已保存的路径时

- 如果该路径处于隐藏状态，那么在图像中使用钢笔工具创建一个新路径会创建一个新的工作路径，但原路径依然存在。
- 如果该路径不处于隐藏状态，此时在图像中使用钢笔工具创建一个新路径相当于继续编辑当前的工作路径，也就是说相当于创建子路径。

另外，可以直接在【路径】面板中创建新的子路径。

3.4　文字工具

Photoshop CS6 中提供的文字功能非常强大，用户可以直接在图像中输入、编辑和修改文字，并对文字进行对齐、排列、间距调整、缩放、颜色调整、拼写检查、文本查找及替换等操作，还可以对文字进行各种特殊变形，使文字产生奇异的形态效果。

3.4.1　创建文字图像和选区

1. 创建点文字

在 Photoshop 中，文字的创建和编辑与其他图像的创建和编辑相比步骤有所不同。在使用工具箱中的大多数工具时，必须先设置好工具的选项，然后创建相应的图像，而使用文字工具创建文字时，可以在创建文字前设置格式，也可以在完成创建后重新设置文字格式。

在 Photoshop 中可以直接沿直线创建文字，也可以沿指定的路径创建文字，使用这两种方法创建的文字称为点文字。

1）直接沿直线创建文字

选择工具箱中的横排文字工具或竖排文字工具，在图像窗口中单击，即可从鼠标单击的位置开始输入文字。文字的输入、编辑和修改方法与 Word、WPS 等软件中的文字编辑方法一样，文字使用的格式为输入前选项栏中设定的格式。如果要对文字格式进行修改，必须先选中要修改的文字，然后在选项栏中进行修改。

2）沿指定的路径创建文字

在 Photoshop 中沿路径创建文字的基本步骤如下：

（1）在图像中创建路径。

（2）选择工具箱中的文字工具，将鼠标指针移动到路径上，当鼠标指针带“～”形状时单击，从鼠标指针落点处开始输入文字，在文字开始的位置会出现一个“O”。

2. 创建文字选区

选择工具箱中的文字蒙版工具，在图像窗口中单击，图像暂时转换为快速蒙版模式，此时在图像中输入文字，实际上是在编辑快速蒙版。文字编辑完成后，单击选项栏中的【确认】按钮，图像将回到标准编辑模式，刚编辑的文字将会转为选区。

在使用文字蒙版工具建立选区后，无法再对文字进行编辑，所以在回到标准编辑模式前一定要确认需要的选区效果已经完成。

3.4.2　创建段落文字

段落文字就是创建在文字定界框内的文字，文字定界框的作用是通过文字工具在图像中划出一个矩形范围，在该范围内创建文字和文字段落，使用文字定界框可以限定图像中文字出现的范围和位置。

1. 利用任意大小的文字定界框创建段落

选择工具箱中的文字工具，在图像中拖曳鼠标指针，图像中将会生成一个文字定界框，随后输入的文字将在定界框内自动换行显示。

如果输入的文字过多，超出定界框范围的文字会被隐藏。

2. 利用自定大小的文字定界框创建段落

按住 Alt 键，在图像中拖曳鼠标指针，在弹出的【段落文字大小】对话框中可以设置文字定界框的大小。

3. 调整文字定界框

- 按住 Ctrl 建，可以移动文字定界框和【定界框中心】标志“⌖”的位置。
- 按住 Ctrl 键，用鼠标拖曳文字定界框的边线可以对定界框进行斜切变形。
- 将鼠标指针移至文字定界框的边线外侧，当鼠标指针显示为⟷时拖曳，以“⌖”(定界框中心)标志为轴旋转定界框。
- 将鼠标指针移至文字定界框的节点上，当鼠标指针显示为双箭头时拖曳，可以调整文字定界框的宽度和高度。

在定界框内的文字显示随定界框形态的改变自动调整。

3.4.3 设置文字属性

在 Photoshop 的工具箱中可以选择不同的文字工具，这一步操作一定要在创建文字前进行。

1. 文字属性

在工具箱中选择文字工具，其选项栏如图 3-41 所示。



图 3-41 文字工具选项栏

1）转换文字方向

单击【文字转换】按钮，可以将水平文字转换为垂直文字，或将垂直文字转换为水平文字。

2）设置文字的字符格式

在文字工具选项栏中可以直接设置字符格式。

- 【字体】：设置文字所要使用的字体。
- 【字形】：设置文字的倾斜、加粗等形态。
- 【字体大小】：设置文字的大小。
- 【锯齿】：选择文字边缘的平滑方式。

3）设置文字的对齐方式

- 在选项栏中选择横排文字工具，【对齐】按钮显示为使水平文字【向左对齐】、【沿水平



中心对齐】和【向右对齐】。

- 在选项栏中选择竖排文字工具，【对齐】按钮显示为使垂直文字【向上对齐】、【沿垂直中心对齐】和【向下对齐】。

4）设置文字颜色

单击【文本颜色】色块可以修改被选中文字的颜色。

5）设置文字变形

在图像中创建了文字后，单击选项栏中的【创建文字变形】按钮，在弹出的【变形文字】对话框中单击【样式】下拉列表，可以对文字进行变形。

2.【字符】面板

在工具箱中选择文字工具，单击选项栏中的【显示/隐藏字符和段落面板】按钮，在弹出的面板组中选择【字符】，如图 3-42 所示。

【字符】面板中包括字体、字体大小、垂直缩放、所选字符的比例间距、基线偏移等设置；【颜色】色块下方有一排按钮用来设置文字字符的效果，当在文字图层中选中要设置的文字时，分别单击这些按钮，可以完成相应的设置。

3.【段落】面板

在工具箱中选择文字工具，单击选项栏中的【显示/隐藏字符和段落面板】按钮，在弹出的面板组中选择【段落】，如图 3-43 所示。



图 3-42　【字符】面板



图 3-43　【段落】面板

【段落】面板主要用于设置文字段落的格式，其中包括对齐方式、缩进方式、段前添加空格、段后添加空格等。

3.5　其他工具

这些工具包括裁剪、切片等辅助工具，它们是对图像进行修改大小、制作切片及在编辑修改过程中进行帮助的工具。

3.5.1 裁剪工具

1. 裁剪工具

裁剪工具主要用来将图像中多余的部分剪切掉，只保留需要的部分；在裁剪的同时还可以对图像进行旋转、扭曲等变形修改，并可直接设置裁剪后图像的像素和分辨率。

在工具箱中选择裁剪工具后，选项栏如图 3-44 所示。



图 3-44 裁剪工具选项栏

- 单击工具选项栏左边的下拉按钮，可以打开工具预设选取器，在预设选取器里可以选择预设的参数对图像进行裁剪。
- 【裁剪比例】：单击该按钮可以显示当前的裁剪比例或设置新的裁剪比例。如果图像中有选区，则按钮显示为选区。
- 【裁剪输入框】：可以自由设置裁剪的长宽比。
- 【纵向与横向旋转裁剪框】：设置裁剪框为纵向裁剪或横向裁剪。
- 【拉直】：可以校正倾斜的照片。
- 【视图】：可以设置 Photoshop CS6 裁剪框的视图形式。
- 其他裁剪选项：可以设置裁剪的显示区域，以及裁剪屏蔽的颜色、不透明度等。
- 【删除裁剪的像素】：勾选该复选框后，裁剪完毕的图像将不可更改；若不勾选该复选框，即使裁剪完毕，选择裁剪工具单击图像区域仍可显示裁剪前的状态，并且可以重新调整裁剪框。

2. 透视裁剪工具

透视裁剪工具是 Photoshop CS6 新增的工具，该工具可以在裁剪的同时方便地校正图像的透视错误，即对倾斜的图片进行校正，其选项栏如图 3-45 所示。

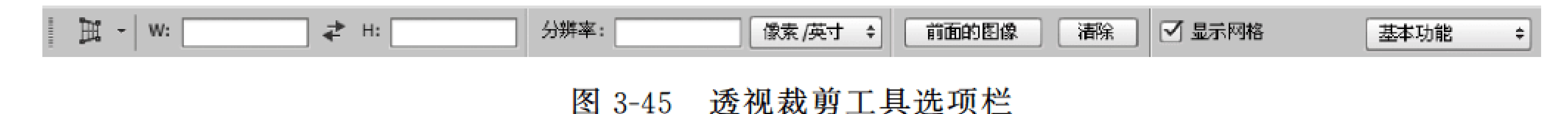


图 3-45 透视裁剪工具选项栏

- 参数输入框：在框中可以输入需要裁剪的尺寸。
- 单位：单击该按钮可以设置裁剪后图像的单位。
- 【前面的图像】：单击该按钮可以使裁剪后的图像与之前打开的图像大小相同。
- 【清除】：单击该按钮可以清除输入框中的数值。
- 【显示网格】：若勾选【显示网格】复选框，可显示裁剪框的网格；若不勾选，则仅显示外框线。

3. 切片工具

Photoshop 加强了对网络的支持，切片工具和切片选择工具就是特别针对网络应用开发的，使用它们可以将较大的图像切割为几个小图像，便于在网上发布，提高网页打开的

速度。

1）创建切片

创建切片的方式通常有两种，一种是根据参考线切片，这种切片方式比较精确；另一种是利用切片工具直接在图像中拖曳鼠标指针切片，这种方式比较灵活，但切片的位置和排列不够精确。

2）切片工具选项栏

选择工具箱中的切片工具，选项栏如图3-46所示。

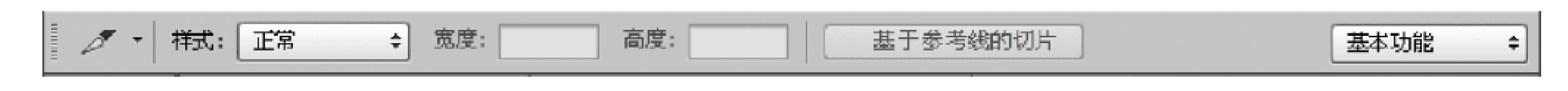


图3-46 切片工具选项栏

- 【样式】下拉列表中有3个选项，选择不同的选项可以在图像中建立大小与比例不同的切片。
- 【基于参考线的切片】按钮只有在图像窗口中设置了参考线时才可使用，单击该按钮可以根据当前图像中的参考线创建切片。

4. 切片选择工具

切片选择工具主要用于对切片进行各种设置，例如切片的堆叠、选项设置、激活、分割、显示或隐藏等。切片选择工具选项栏如图3-47所示。

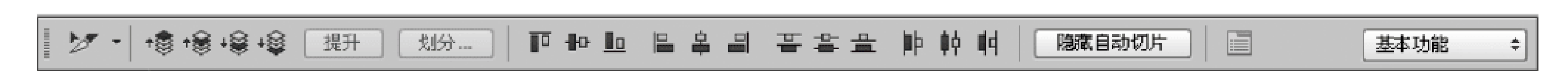


图3-47 切片选择工具选项栏

1）显示/隐藏切片

在创建切片的时候，如果从图像的左上角开始创建切片，切片左上角默认的编号显示"01"；如果从其他位置开始创建，则新创建切片的编号可能是"02"（从沿一边中间位置开始创建切片）或"03"（从图像的中间位置开始创建切片），这是因为当不是从左上角开始创建切片时，系统根据创建切片的边线将图像的其他部分自动分割，变成了一些自动切片，切片的默认编号是从左上角开始的，系统将会对所有切片（包括隐藏切片）进行编号。

当在图像中创建切片时，只有拖曳鼠标指针生成的切片是被激活的，其他自动产生的切片都是自动切片，自动切片默认是隐藏的。单击【显示自动切片】按钮，即可将自动切片显示出来，此时【显示自动切片】按钮将显示为【隐藏自动切片】按钮，自动切片的边线显示为虚线，此时再单击选项栏中的【隐藏自动切片】按钮，即可将自动切片再次隐藏起来。

2）调整切片的大小

选择切片选择工具，将鼠标指针移动到当前切片的边线或四角上，当鼠标指针显示为双箭头时拖曳，可以通过移动切片边线的位置来调整切片的大小，按Delete键可以删除当前切片。

3）激活切片

当自动切片左上角的编号显示为灰色时，表示该切片没有被激活，此时切片的部分功能不能使用。如果要使这些切片功能可以使用，只要在图像窗口中单击要激活的切片将其选择，再单击选项栏中的【提升】按钮就可以将其激活，此时切片左上角的编号显示为蓝色。系

统默认被选择切片的边线显示为橙色,其他切片的边线显示为蓝色。

4) 设置切片的堆叠顺序

切片像图层一样,每个切片之间有一定的堆叠顺序,选项栏左侧的 4 个按钮就是用来设置切片堆叠顺序的。在图像中选择要设置堆叠顺序的切片,可以分别单击上述按钮完成相应操作。

5) 平均分割切片

用户可以将现有的切片进行平均分割。选择切片选择工具后,在图像窗口中选择一个切片,单击选项栏中的【划分】按钮,将弹出【划分切片】对话框,如图 3-48 所示。

(1) 勾选【水平划分为】复选框,可以通过添加水平分割线将当前切片在高度上进行分割。

(2) 勾选【垂直划分为】复选框,可以通过添加垂直分割线将当前切片在宽度上进行分割。



图 3-48 【划分切片】对话框

6) 将切片的图像保存为网页

在 Photoshop 中处理图像切片,最终目的是在网上发布,因此要把它们保存为网页的格式。Photoshop 提供了最佳的处理网页图像文件的工具和办法,选择菜单栏中的【文件】|【存储为 Web 所用格式】命令,在弹出的【存储为 Web 所用格式】对话框中进行设置,单击【存储】按钮,将弹出【将优化结果存储为】对话框,在对各选项进行相应设置后单击【保存】按钮,即可进行保存。

3.5.2 辅助工具

1. 吸管工具

吸管工具用于快速、准确地取样颜色。使用吸管工具可以从图像中吸取某一点的颜色,或者以拾取点周围的平均色进行颜色取样,从而改变前景色,其选项栏如图 3-49 所示。



图 3-49 吸管工具选项栏

- 【取样大小】:单击选项栏中的【取样大小】选项的下三角按钮,可弹出下拉菜单,在其中可选择要在怎样的范围内吸取颜色。
- 【样本】:对选取的颜色在哪个层进行设置。
- 【显示取样环】:勾选【显示取样环】复选框,则在 Photoshop CS6 图像中单击取样点时出现取样环。

在图像中要吸取的颜色上单击,工具箱中的【前景色】色块将被修改为吸管工具吸取的颜色;按住 Alt 键不放单击,使用吸管工具吸取的颜色将被用作背景色。

2. 颜色取样器工具

颜色取样器工具的主要功能是检测图像中像素的色彩构成。在图像中单击,鼠标指针

落点处会出现一个色彩样例的图标，称为取样点。在同一个图像中最多可以放置 4 个取样点，每个取样点处像素的色彩构成都显示在【信息】面板中。

选择颜色取样器工具，并打开一幅图像，在图像中单击 4 次，每单击一次，在图像中便会出现一个取样点图标，其右下角分别标注有“1”、“2”、“3”、“4”，表示它们分别是“＃1”、“＃2”、“＃3”、“＃4”取样点，此时【信息】面板下方显示每个取样点的色彩构成。【信息】面板如图 3-50 所示。



图 3-50 【信息】面板

将鼠标指针移动至取样点的位置，当鼠标指针变成箭头形状时拖曳，可以调整取样点的位置。按住 Alt 键不放，移动鼠标指针至取样点的位置，当鼠标指针变成剪刀的形状时单击，可以删除该取样点。图 3-51 所示为颜色取样器工具选项栏。



图 3-51 颜色取样器工具选项栏

3. 标尺工具

标尺工具的主要功能是对某部分图像的长度或角度进行精确的测量，测量的数据显示在标尺工具选项栏和【信息】面板中。

选择工具箱中的标尺工具，然后在图像上需要测量的起点按住鼠标左键并向终点拖动，到达终点后松开鼠标，即可完成这两点之间的距离测量。标尺工具选项栏如图 3-52 所示。



图 3-52 标尺工具选项栏

其中分别是起点坐标、宽度、高度、角度和距离。

4. 附注工具

附注工具是在图片中添加注释的工具。其使用起来比较简单，选择这款工具后在需要添加注释的地方单击一下就会有一个对话框出现，在里面输入想要的文字即可，关闭对话框就可以保存。用户可以在选项栏中输入其他信息，也可以在注释上右击选择【删除】等命令。这款工具可以多次使用，注释完成后，保存为. psd 格式的文件就可以把注释保存。选择附注工具后的选项栏如图 3-53 所示。

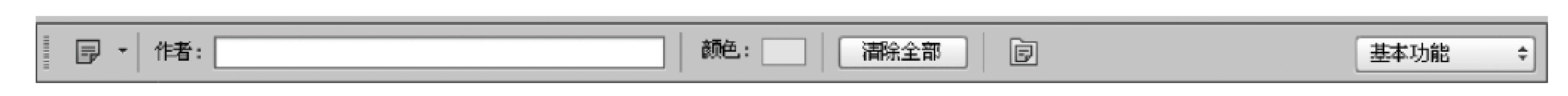


图 3-53 附注工具选项栏

- 【作者】文本框：添加作者名称，它会显示在【文本附注】框的标题栏中。
- 【颜色】色块：显示图像中的附注图标所使用的颜色，单击该色块，可以在弹出的对话框中对此颜色进行调整。

- 【清除全部】按钮：可以将图像中所有的文本附注删除。
- 【显示/隐藏注释面板】按钮：用来显示/隐藏注释面板。

当在图像中添加了文本附注或语音附注后，如果要在保存图像时就将这些附注保存，则只能将文件保存为.psd、.pdf或.tif格式，并且保存时要在【存储为】对话框中勾选【附注】复选框。

5. 计数工具

计数工具可用来统计图像中对象的个数，并将数目显示在选项栏中，其选项栏如图3-54所示。

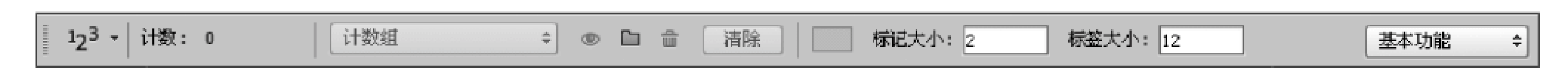


图3-54　计数工具选项栏

- 【显示计数的总个数】：对象的个数。
- 【计数组】：显示计数组的名字及重新命名。
- 【切换计数组的可见性】：设置计数组的可见性。
- 【创建新的计数组】：创建新的计数组。
- 【删除计数组】：将当前计数组删除。
- 【清除】：删除计数。
- 【标签颜色】：设置计数标签颜色。
- 【标记大小】：设置计数标记大小。
- 【标签大小】：设置计数标签大小。

3.6　综合应用

3.6.1　奥运五环的制作

制作奥运五环的步骤如下：

(1) 选择菜单栏中的【文件】|【新建】命令或按快捷键Ctrl+N，在打开的图3-55所示的【新建】对话框中设置各参数，然后单击【确定】按钮，创建一个图形文件。

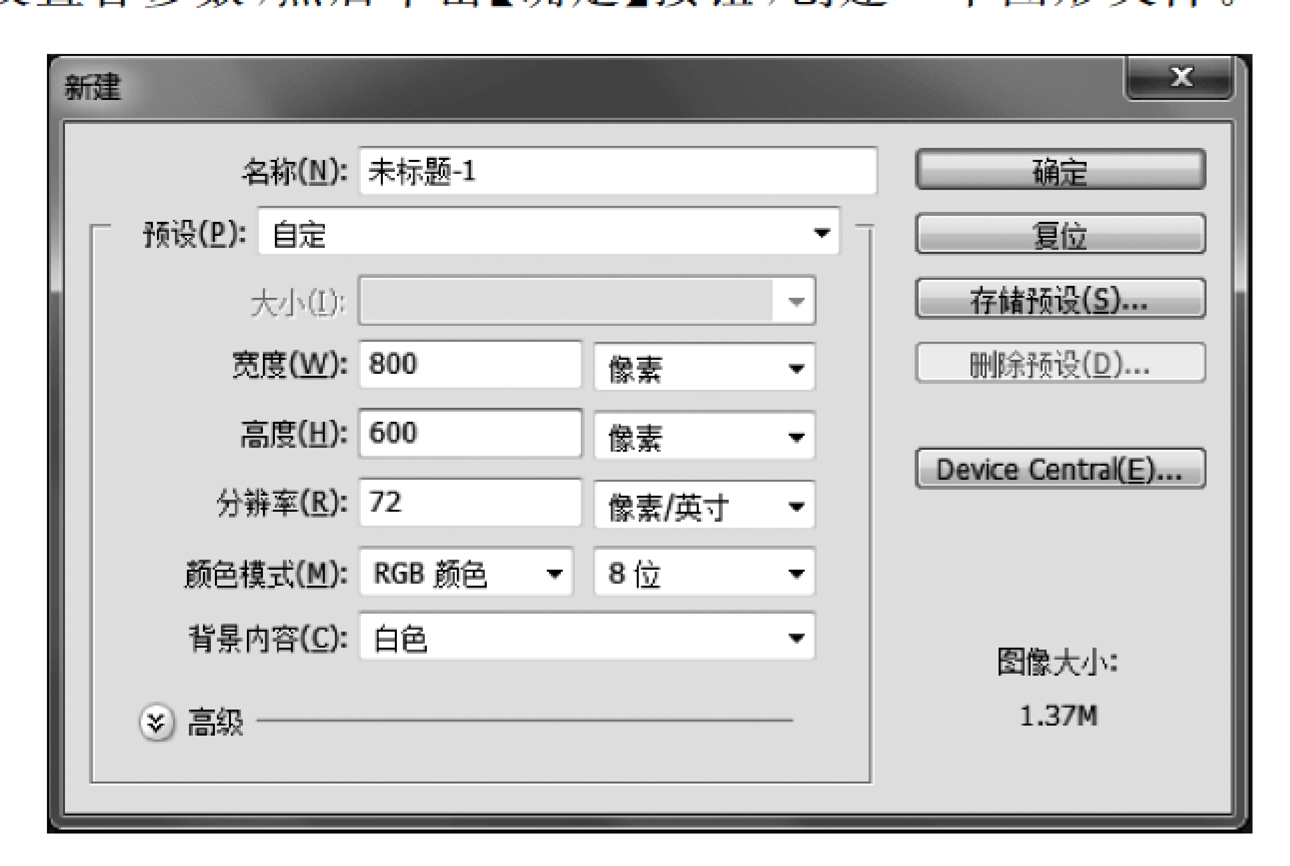


图3-55　【新建】对话框

(2) 在【图层】面板上单击右边第2个按钮(将鼠标指针放在上面几秒会看到提示：创建新图层)，新建一个透明图层，如图3-56所示。

(3) 选择工具箱里的椭圆选框工具，在按住Shift键的同时拖曳(Shift键的作用是拖曳出正圆选区，如果想画出正方形选区，在选择矩形选框工具的同时按住Shift键)，创建一个正圆形选区，如图3-57所示。

(4) 给正圆选区填充颜色。在色板里选择红色进行填充，方法是按Alt+Delete(注意，快捷键Alt+Delete是填充前景色的，如果想填充背景色，可以用Ctrl+Delete快捷键)，填充后的效果如图3-58所示。



图3-56　创建新图层

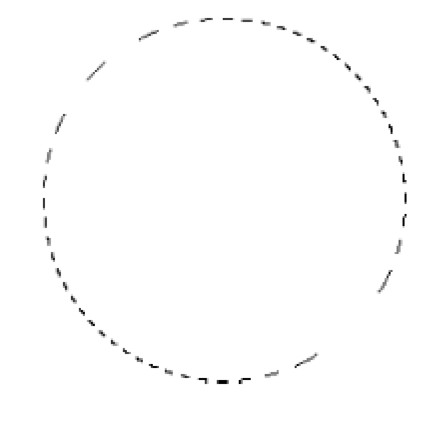


图3-57　创建正圆选区

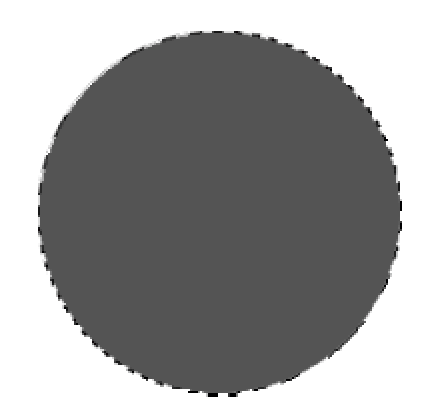


图3-58　填充颜色

(5) 在选中的圆上单击鼠标右键，选择【变换选区】命令，然后在按住Alt+Shift键的同时从对角线的地方缩小选区，这个快捷键的作用是让选区等比例缩小，感觉大小合适后按Enter键确定，再按Backspace键删除其中的部分，效果如图3-59所示，接着按Ctrl+D键取消选区。

(6) 按Ctrl+J键4次，复制出4个圆环，快捷键Ctrl+J用于复制图层。然后在按住Ctrl键的同时分别单击图层的缩略图选择每个图层，选择前景色分别为黄、绿、黑、蓝，为图层填上颜色，并为图层重新命名。复制后的效果如图3-60所示。

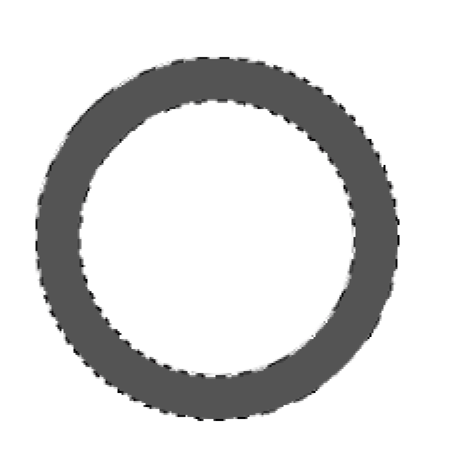


图3-59　圆环效果

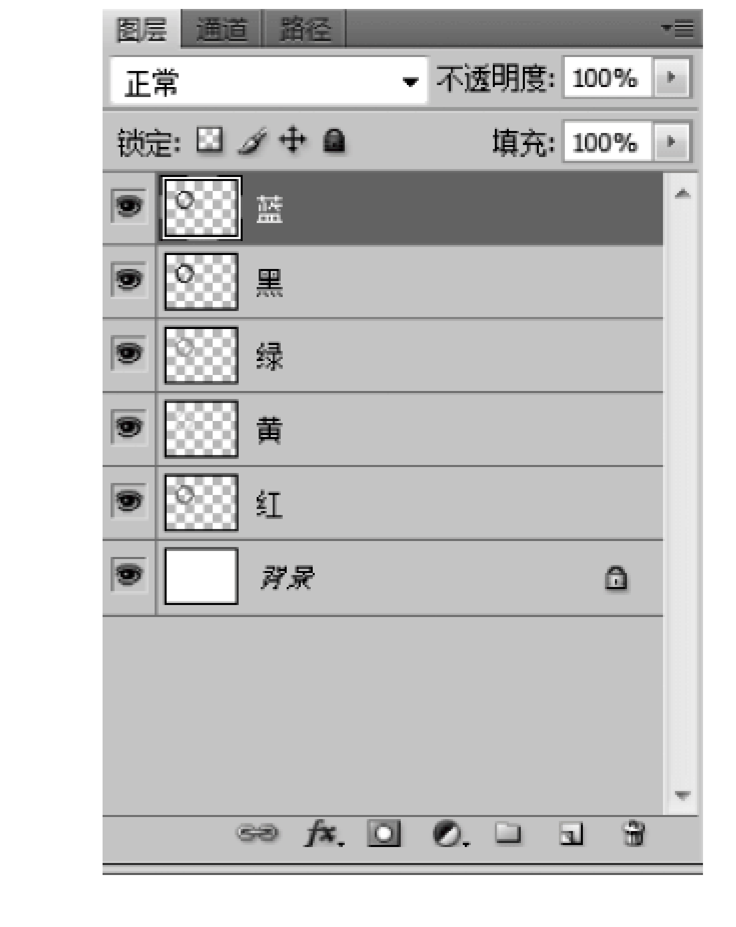


图3-60　复制图层

(7) 选择工具箱里的移动工具(第1个),把每个圆环移动到合适的位置,效果如图3-61所示。在移动的时候可以按住Shift键移动,这样能保证每个圆环在水平位置上移动。

(8) 选中红色图层,在按住Ctrl键的同时单击红色图层的缩略图,然后选择工具箱中的橡皮擦工具,单击【图层】面板里的黑色图层,用橡皮擦擦除红色与黑色相交的部分(有两部分相交,只需擦除其中的一部分,这样才会有黑色从红色环中穿插的效果)。接着用同样的方法选择黄色选区,在黑色图层上擦除,在蓝色图层上擦除,再选择绿色图层,在蓝色图层上擦除,最终效果如图3-62所示。

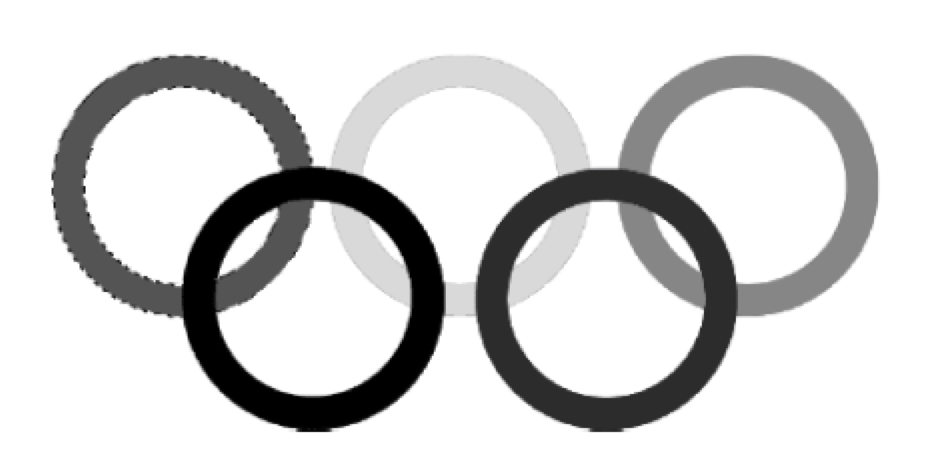


图3-61 移动圆环

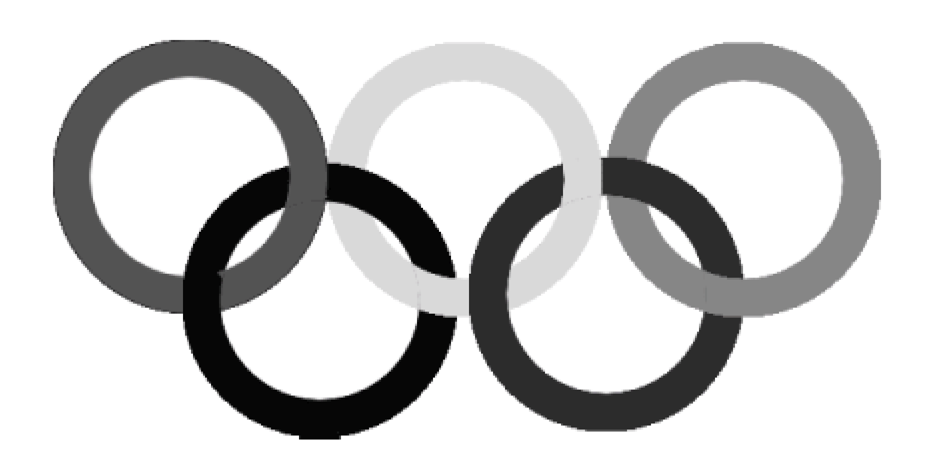


图3-62 奥运五环效果图

3.6.2 矢量图案的制作

其制作步骤如下:

(1) 选择菜单栏中的【文件】|【新建】命令,创建一个图形文件。然后选择椭圆工具,绘制图3-63所示的路径。

(2) 选择路径选择工具,选中绘制好的椭圆路径,按Ctrl+C键复制路径,再按Ctrl+V键粘贴路径。选择复制后的路径,按Ctrl+T键对路径进行变形,再按住Alt键将复制后的椭圆水平放大一些,变形后的效果如图3-64所示。

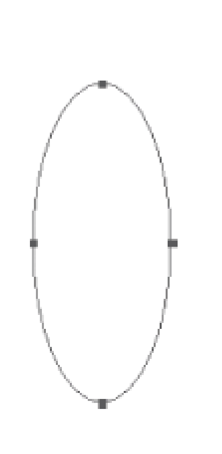


图3-63 绘制椭圆路径

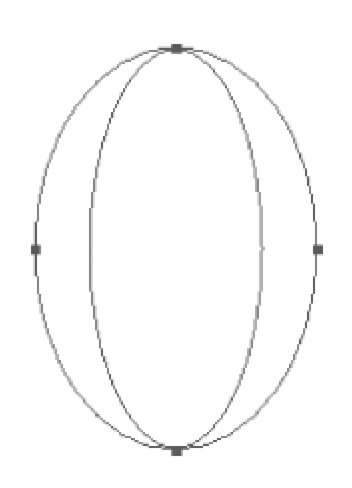


图3-64 路径的复制和变形

(3) 选择添加锚点工具,移动鼠标指针到椭圆右侧如图3-65所示的位置,在椭圆路径上添加两个锚点。

(4) 选择直接选择工具,选择椭圆右侧的锚点,按Delete键删除选择的锚点,删除后的效果如图3-66所示。

(5) 选择钢笔工具,移动鼠标指针到椭圆右侧的锚点上,绘制人脸的侧面图形,效果如图3-67所示。

(6) 新建图层并选中,将前景色调为黑色,再选择路径选择工具,选中所有路径,单击选项栏中的【排除重叠区域】按钮,显示【路径】面板,然后单击【前景色填充】按钮用前景色进行填充,如图3-68所示。

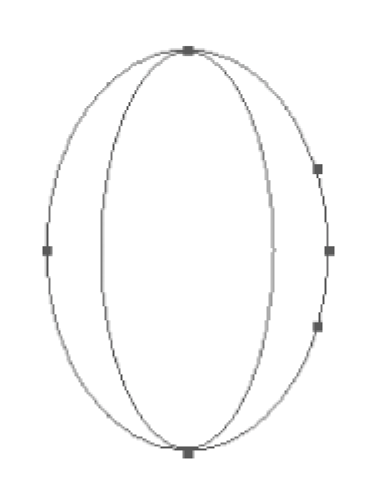


图 3-65 添加锚点

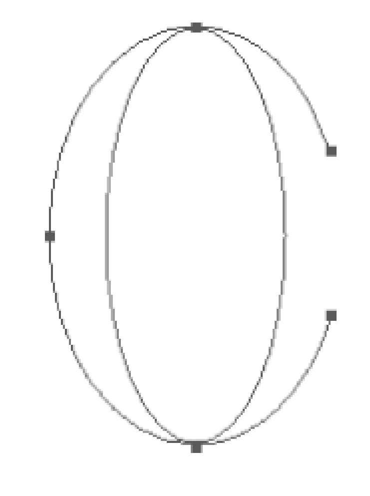


图 3-66 删除锚点

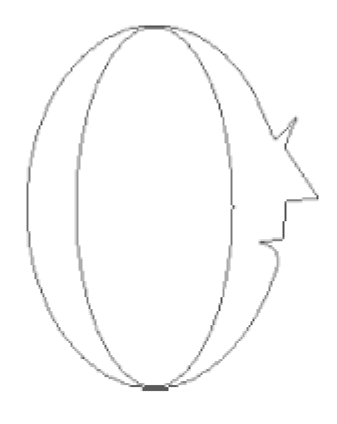


图 3-67 用钢笔工具绘制后的图形

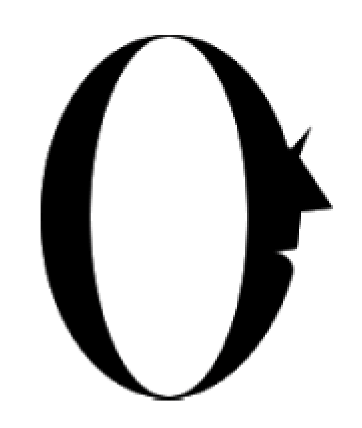


图 3-68 填充颜色

(7) 选择矩形工具，在图案左侧位置拖出一个矩形区域，按 Delete 键删除，效果如图 3-69 所示。

(8) 按 Ctrl+J 键复制图层，将填充色改为灰色，然后按 Ctrl+T 键进行自由变换，选择【编辑】|【变换】|【水平翻转】命令，得到图 3-70 所示的效果，最后将文件保存。

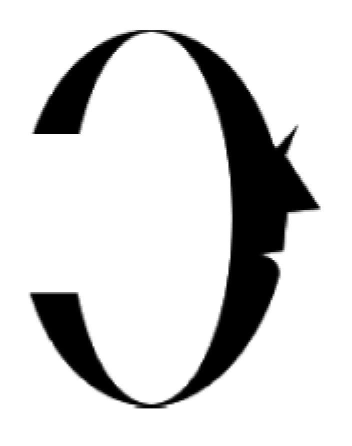


图 3-69 删除部分区域



图 3-70 复制图形并翻转

3.6.3 图像的修剪

修剪图像的操作步骤如下：

(1) 选择菜单栏中的【文件】|【打开】命令，打开“素材 1”文件。

(2) 选择工具箱中的裁剪工具，图形中心出现中心点、四周出现滚动线条，在图像上单击，图像上会出现网格，效果如图 3-71 所示。

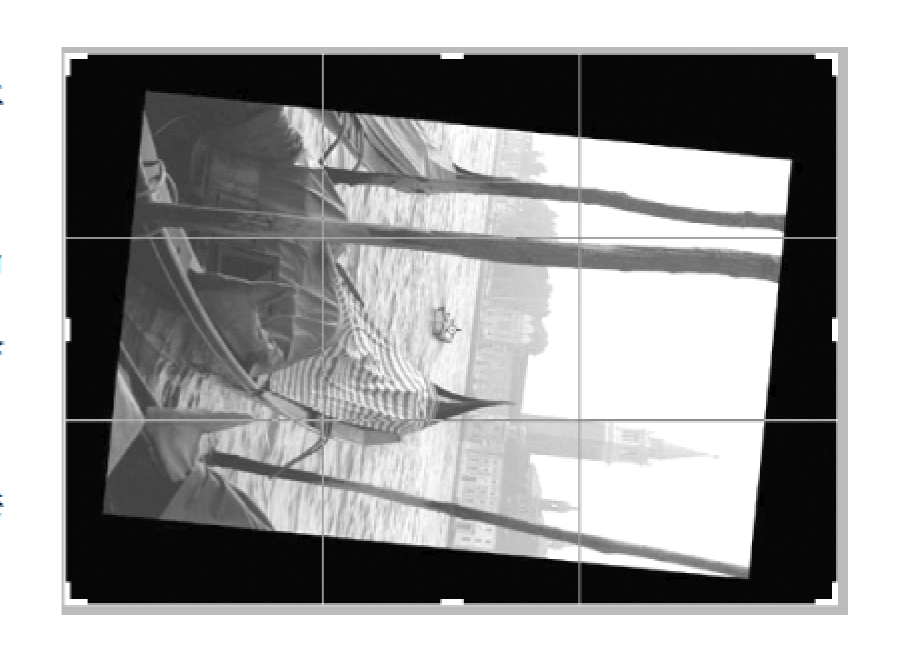


图 3-71 选择裁剪工具并单击时图像的显示

(3) 此时将鼠标指针移至图像的右上顶角，旋转图像，让网格与图像方向一致，效果如图 3-72 所示。

(4) 调整网格大小，让网格与要留下的图像大小一致，效果如图 3-73 所示。



图 3-72 旋转图像



图 3-73 调整网格大小

(5) 单击选项栏上的【确认】按钮确定剪裁结束,效果如图 3-74 所示。

(6) 选择菜单栏中的【图像】|【图像旋转】|【180 度】命令,效果如图 3-75 所示。



图 3-74 剪裁后的图像



图 3-75 旋转图像

(7) 保存文件,图像的修整结束。

3.6.4 文字标识的制作

(1) 选择菜单栏中的【文件】|【新建】命令,创建一个新的图形文件,将宽和高分别设置为 500 像素和 200 像素。

(2) 选择工具箱中的横排文字工具,在画布上的某位置单击并输入内容"Tsnow",然后调整字体、字号,效果如图 3-76 所示。

Tsnow

图 3-76　添加并修改文字

（3）选中文字图层，然后右击，在弹出的快捷菜单中选择【创建工作路径】命令，将文字图层隐藏。接着打开【路径】面板，查看文字工作路径，效果如图 3-77 所示。



图 3-77　文字工作路径

（4）选择路径选择工具，将 T 移到 s 上。然后选择直接选择工具，选中 T 右侧的两个锚点，按住 Shift 键，以直线方式将其拖至 w 的边缘，效果如图 3-78 所示。

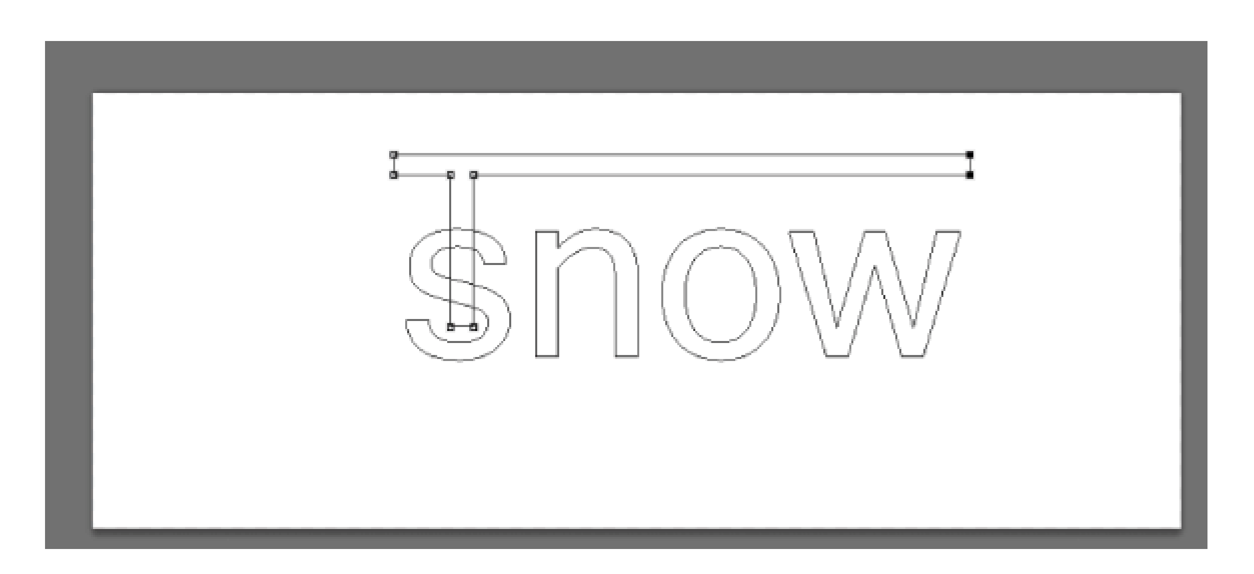


图 3-78　文字工作路径的修改

（5）选择矩形工具，在文字“now”中部加一个矩形路径。然后选择直线工具，按住 Shift 键在矩形下面加一条 3 像素的直线路径，效果如图 3-79 所示。

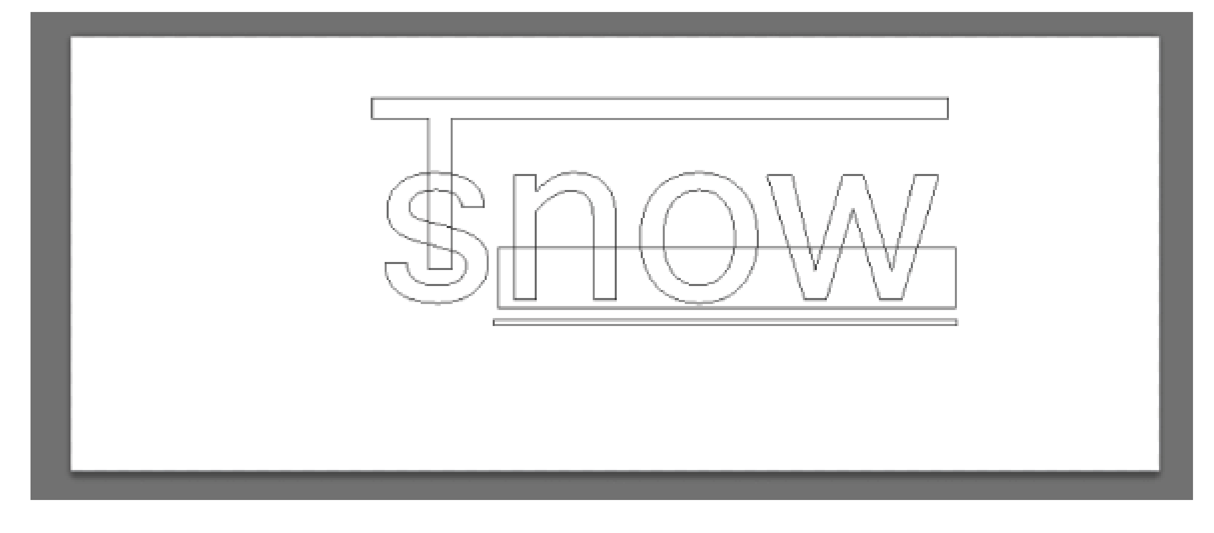


图 3-79　添加路径

(6) 用路径选择工具选中所有路径,单击选项栏中的【路径操作】按钮,选择【排除重叠形状】选项,效果如图 3-80 所示。

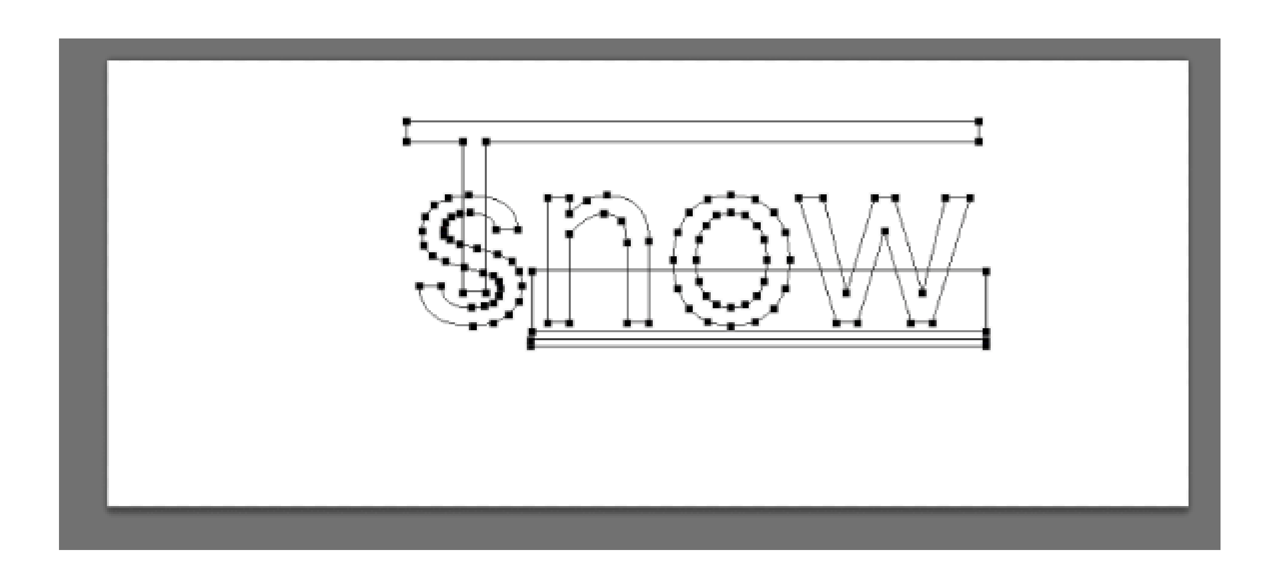


图 3-80 排除重叠形状

(7) 在【图层】面板中新建一个图层并选中,将前景色设为深红色。然后返回【路径】面板,单击【路径】面板下方的【前景色填充】按钮,效果如图 3-81 所示。



图 3-81 填充颜色

(8) 用户可以根据需要设置图层样式等效果,使其变得更加立体,最后保存文件。

3.6.5 Logo 的制作

(1) 选择菜单栏中的【文件】|【新建】命令,创建一个新的图形文件,将宽和高均设置为 500 像素,背景为白色。然后创建新层,用钢笔工具画出路径,在【图层】面板中单击【新建图层】按钮,新建"图层 1",设置前景色为#97d623。接着返回【路径】面板,单击【前景色填充】按钮填充颜色,效果如图 3-82 所示。

(2) 使用加深、减淡工具涂抹,将其加深和减淡,做出立体效果,如图 3-83 所示。

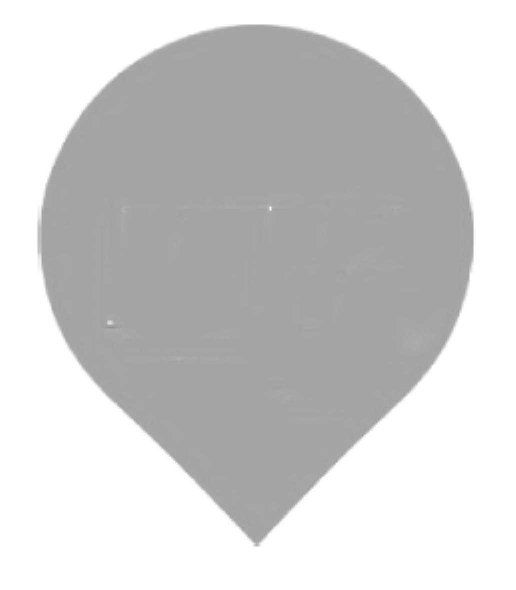


图 3-82 绘制图形并填充

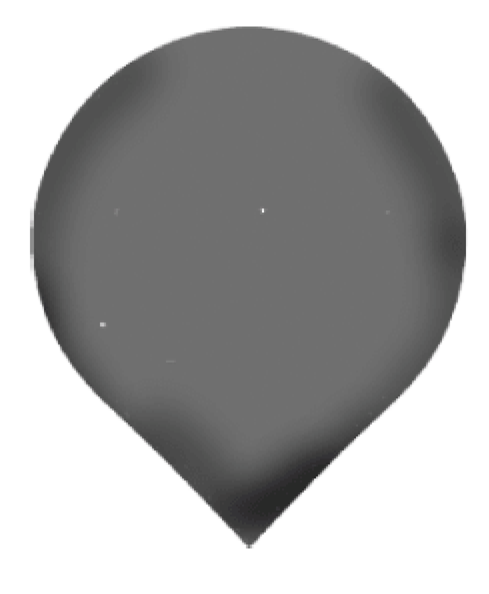


图 3-83 制作立体效果

（3）新建“图层 2”，绘制一个正圆选区，填充颜色＃e6e3dc，然后用加深工具加暗，效果如图 3-84 所示。

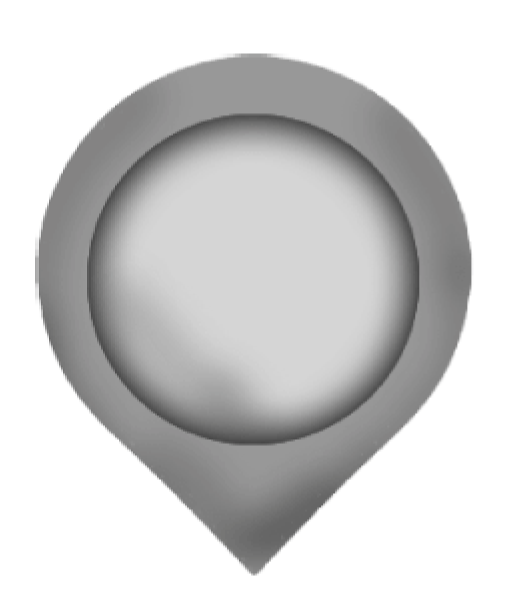


图 3-84　新图层效果

（4）单击【图层】面板下方的【图层样式】按钮，为图层添加样式，样式参数如图 3-85 所示。



图 3-85　图层样式参数

（5）新建“图层 3”，在自选图形中选择音符，拖动绘制图 3-86 所示的形状，颜色为＃94d718。然后添加图层样式，样式参数如图 3-87 所示。



图 3-86　添加自选图形

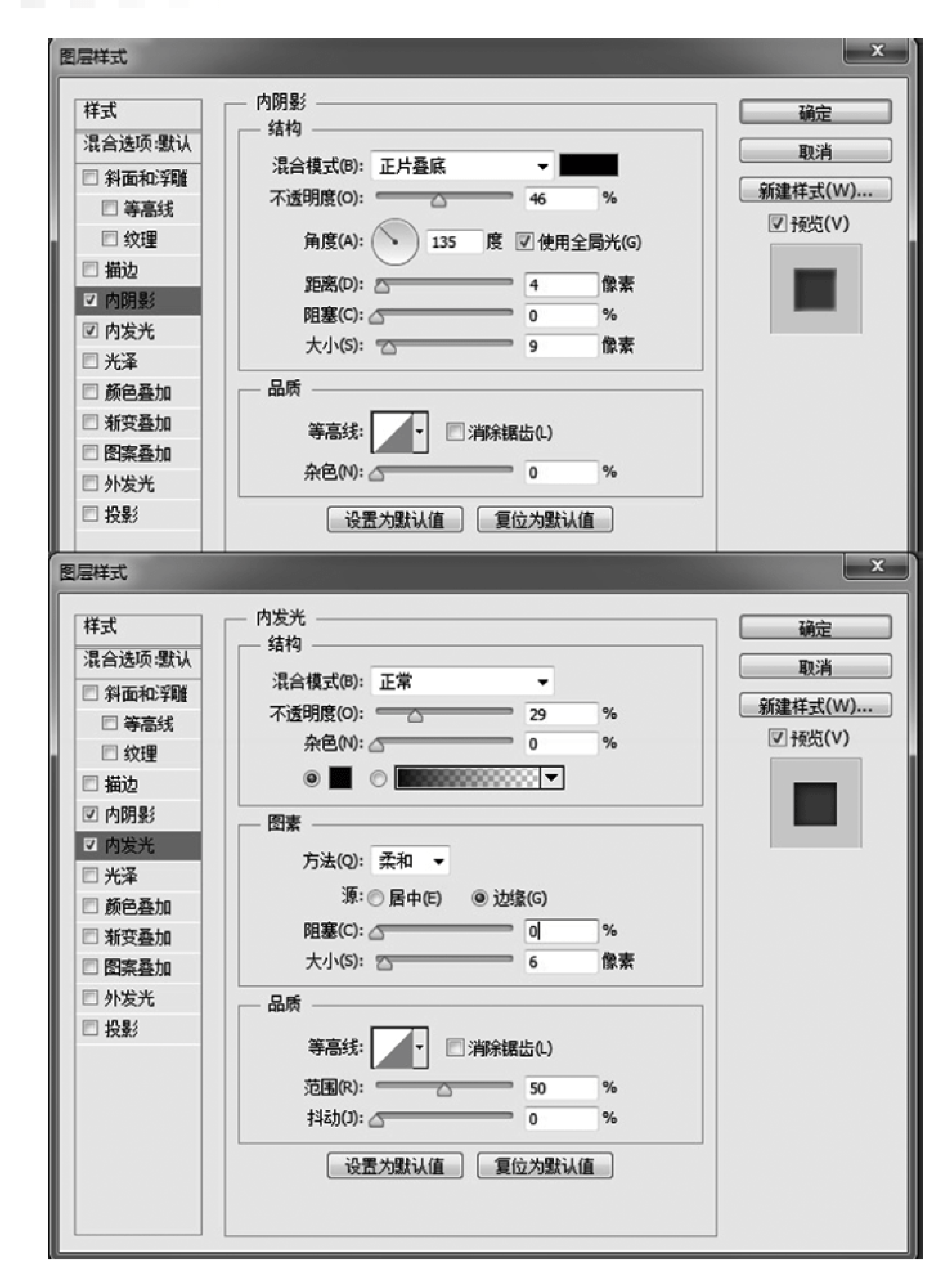


图 3-87 自选图形样式参数

(6) 新建“图层 4”,调出符号选区,填充白色,图层不透明度为 50%。然后用套索工具删除,效果如图 3-88 所示。

(7) 新建“图层 5”,羽化值为 5,用椭圆工具绘制一个小的椭圆,为整体添加阴影,最终效果如图 3-89 所示。



图 3-88 图层 4 的修改效果



图 3-89 最终效果图

第4章 Photoshop图层、通道和蒙版

本章学习目标：

✍了解图层、通道、蒙版的基础知识。
✍掌握图层、通道、蒙版的各种操作。
✍掌握图层、通道、蒙版的综合应用。

4.1 图层

图层是 Photoshop 中最常用的容器，我们建议将不同的对象放置于不同的图层上。

4.1.1 图层的基本概念

什么是图层？为了方便读者理解，可以打一个简单的比方来说明。

比如要在一张纸上绘画，当需要在画上添加一些新的图案时，可以先在纸上铺一张透明纸，在这张透明纸上再绘制要添加的图案，并可以通过移动纸或透明纸的位置来改变两层图案的相对位置，也可以添加或拿开部分透明纸来观察在图像中添加或减去部分内容后的效果。用户可以根据需要添加很多层透明纸，以便对图像的效果进行灵活调整，实际上这种方法常用于动画片的制作。

Photoshop 就是利用了这种原理，绘制图像的纸和这些透明纸相当于 Photoshop 中的图层，这种图层堆放的关系称为堆叠。一个文件中的所有图层都具有相同的分辨率、相同的通道数以及相同的图像模式。

1. 常用的图层类型

Photoshop 中的图层类型很多，下面首先了解一些经常用到的图层类型。

1）普通层

普通层是最基本的图层类型，它在图像中的作用相当于前面所说的一张透明纸。

2）背景层

在 Photoshop 中，背景层相当于绘画时最下层的不透明纸。一幅图像可以没有背景层，但如果有则只能有一个背景层。背景层无法与其他层交换堆叠次序，但背景层可以与普通层相互转换。

3）文字层

使用工具箱中的文字工具在图像中创建文字后，系统将自动新建一个图层。在【图层】面板中，如果某层的缩览图为“T”图标，则该层为文字层。文字层主要用于编辑文字的内容、属性和方向。文字层可以进行移动、调整堆叠顺序、复制等操作，但大多数编辑工具和命令不能在文字层中使用，如果要使用，需要将文字层转换为普通层。

4）调节层

通过调节层可以调节其下方所有图层中图像的色调、亮度及饱和度等。在【图层】面板中，调节层的图层缩览图会根据调节层的具体类型发生变化。

5）效果层

用户可以对图层使用图层样式，也就是使该层产生立体、发光及填充等效果。当为一个图层应用图层样式时，该层右侧将出现 fx ，表示该图层是带有图层样式的效果层。

6）形状层

形状层是利用工具箱中的图形工具创建的图层，它主要用于在图像中创建各种矢量形状，如矩形、花朵等。该类图层主要包含 3 部分内容，即填充内容、形状和矢量蒙版。

7）图层组

图层组是图层的组合，它的作用相当于 Windows 系统资源管理器中的文件夹，主要用于组织和管理连续图层。

8）蒙版层

图层蒙版的作用是根据蒙版中颜色的变化使其所在层图像的相应位置产生透明效果。蒙版层的内容比较复杂，而且一般需要与其他命令和工具结合起来使用。

9）图层剪贴组

在图层剪贴组中用基底层（基底层是指图层剪贴组中最下方的图层）充当整个组的蒙版。也就是说，一个图层剪贴组的不透明度是由基底层的不透明度来决定的。

10）智能对象

在 Photoshop 中可以通过转换一个或多个图层来创建智能对象。在【图层】面板中双击智能对象符号的图层缩略图，就能创建一个新的文件图像。在对新图像编辑后保存，原文件中的“智能对象”也会自动更新。

2.【图层】面板

【图层】面板是用来管理和操作图层的，对图层进行的大多数设置和修改操作都是在【图层】面板中完成的。打开文件后的【图层】面板如图 4-1 所示。

- 【图层标签】：位于【图层】面板的左上角，单击可将其置为当前工作状态。
- 【面板菜单】按钮：位于面板的右上角，单击可以弹出控制面板的下拉菜单。
- 【图层分类搜索】按钮：在其下拉列表中选择搜索图层的依据，按类型、名称、颜色、效果、模式、属性对不同的图层进行搜索，以将相关的图层显示出来。



图 4-1 【图层】面板

- 【图层混合模式】下拉列表：在下拉列表中选择当前图层与其下面图层混合的模式。
- 【锁定图层】选项组：对图层中的透明度、像素、位置进行锁定。
- 【不透明度】：决定当前层的透明程度。
- 【填充】：主要适用于带样式的图层。在一个带样式的图层中，【不透明度】文本框中的值会同时影响当前图层中图像和样式的透明度，【填充】框中的值只影响当前图层中的图像透明度，不影响样式透明度。
- 【图层缩略图】：显示本层图像的缩略图，它随着图层图像的变化随时更新，便于用户在处理图像时做参考。
- 【图层名称】：显示图层的名称，创建或复制后的新图层会默认指定一个图层名称，为了方便编辑与查询，可以双击图层名称，在文本框中输入自定义的名称。
- 【显示/隐藏图层】图标：用于表示图层处于显示还是隐藏状态。
- 【链接图层】按钮：当选择两个以上的图层时，该按钮才可用。单击该按钮，可以链接两个或多个图层。链接后的图层可以被一起移动，也可以在链接图层间执行对齐与分布、合并等操作。
- 【添加图层样式】按钮：单击该按钮，在弹出的菜单中选择相应的命令，可以在图像中添加投影、发光、浮雕、渐变及图案等效果。这些效果被链接到当前图层的内容上，在移动或编辑图层中的内容时，图层效果被相应更改。这些图层效果常被用于加强图像中的效果。
- 【添加图层蒙版】按钮：单击该按钮，可以在当前层上创建图层蒙版。如果先在图像中创建适当的选区，再单击此按钮，则可以根据选区范围在当前层上建立适当的图层蒙版。
- 【创建新的填充或调整图层】按钮：单击该按钮，可以在当前层上方创建一个新填充图层或调整图层，从而对当前图层下面的图层进行色调、明度等颜色调整。
- 【创建新组】按钮：单击该按钮，在【图层】面板中将创建一个新的组，类似一个文件夹，便于对图层进行管理与查询。
- 【创建新图层】按钮：单击该按钮，可以在当前层上方创建一个新的透明图层。
- 【删除图层】按钮：单击该按钮，可以删除当前图层。

4.1.2 图层的基本操作

1. 选择图层

- 选择单个图层：在需要操作的图层上单击，当图层显示为蓝色时，表示该图层是当前编辑图层。
- 选择多个图层：按住 Ctrl 键，可以间隔选取多个图层，按住 Shift 键则选取两个图层之间的所有图层。

2. 修改图层名称

在图层的名称上双击，即可修改图层的名称。

3. 新建图层

单击【创建新图层】按钮，可以在当前层的上方添加一个新图层，新添加的图层为普通层。

4. 调整图层堆叠位置

常用的调整图层堆叠位置的方法有以下两种：

- 利用鼠标在【图层】面板中直接拖曳调整图层堆叠位置。
- 打开【图层】面板，在要移动的图层上按住鼠标左键不放拖曳至目的位置，释放鼠标即可。

5. 复制图层

拖曳要复制的图层至【创建新图层】按钮上，然后释放鼠标左键，即可在被复制的图层上方复制一个新图层。

6. 删除图层

拖曳要删除的图层至【删除图层】按钮上，可以直接删除该图层。

7. 合并图层

在制作复杂的实例时，可以将不需要进行调整的多个图层合并，以方便后面的操作。合并图层的菜单命令有【合并图层】、【合并可见图层】和【拼合图像】，在【图层】菜单中选择相应的命令，即可在不同情况下合并图层。【图层】菜单如图 4-2 所示。



图 4-2 【图层】菜单

8. 对齐图层

在制作图像的过程中，经常需要将几个图层向左、向右、向上、向下或居中对齐。

1）对齐多个图层

在同时选中了多个图层后，选择菜单栏中的【图层】|【对齐】命令。

2）对齐链接图层

当【图层】面板中有链接图层时，选择链接图层中的某个图层后，利用菜单栏中的【图层】|【对齐】命令也可以对齐图层。此时当前选择的图层会作为基准，即该图层中的图像不动，其他图层与该图层对齐。

3）将图层与选区对齐

当图像中存在选区时，可以将当前图层与选区对齐。在【图层】面板中选中要向选区对齐的图层，然后选择菜单栏中的【图层】|【将图层与选区对齐】命令，接着在弹出的子菜单中选择所需的命令，即可将当前图层与选区按要求对齐。

9. 分布图层

在制作图像的过程中，经常需要将几个图层平均分布，例如制作一排间距相等的栅栏等，Photoshop 提供了平均分布图层的命令。

使用平均分布图层的命令有两个必要条件，一是必须有两个以上的图层，二是这些图层必须全部是链接图层。在分布前，先选择两个图层，将它们移动到分布的起点和终点位置，然后将所有要平均分布的图层链接，最后选择菜单栏中的【图层】|【分布】命令将各层平均分布。

4.1.3　图层样式

在前面介绍图层类型时曾经讲过，在【图层】面板上某层的右侧若出现【效果层】图标，该图层就是一个效果层，在效果层中可以产生立体、发光及填充等效果。效果层实际上是应用了图层样式的图层。

1. 使用图层样式

选择【图层】|【图层样式】|【混合选项】命令，或双击图层缩略图或者图层名称右侧的空白处，或单击【图层】面板底部的【图层样式】按钮，均可弹出【图层样式】对话框，如图 4-3 所示。该对话框左侧为图层样式选项列表区，在该对话框中可以设置多个图层样式；中间为参数设置区，在此可以设置各个图层样式的参数，从而获得不同的图层样式效果；右侧为预览区，用户能够在设置参数时实时预览到参数调整对整体效果的影响。

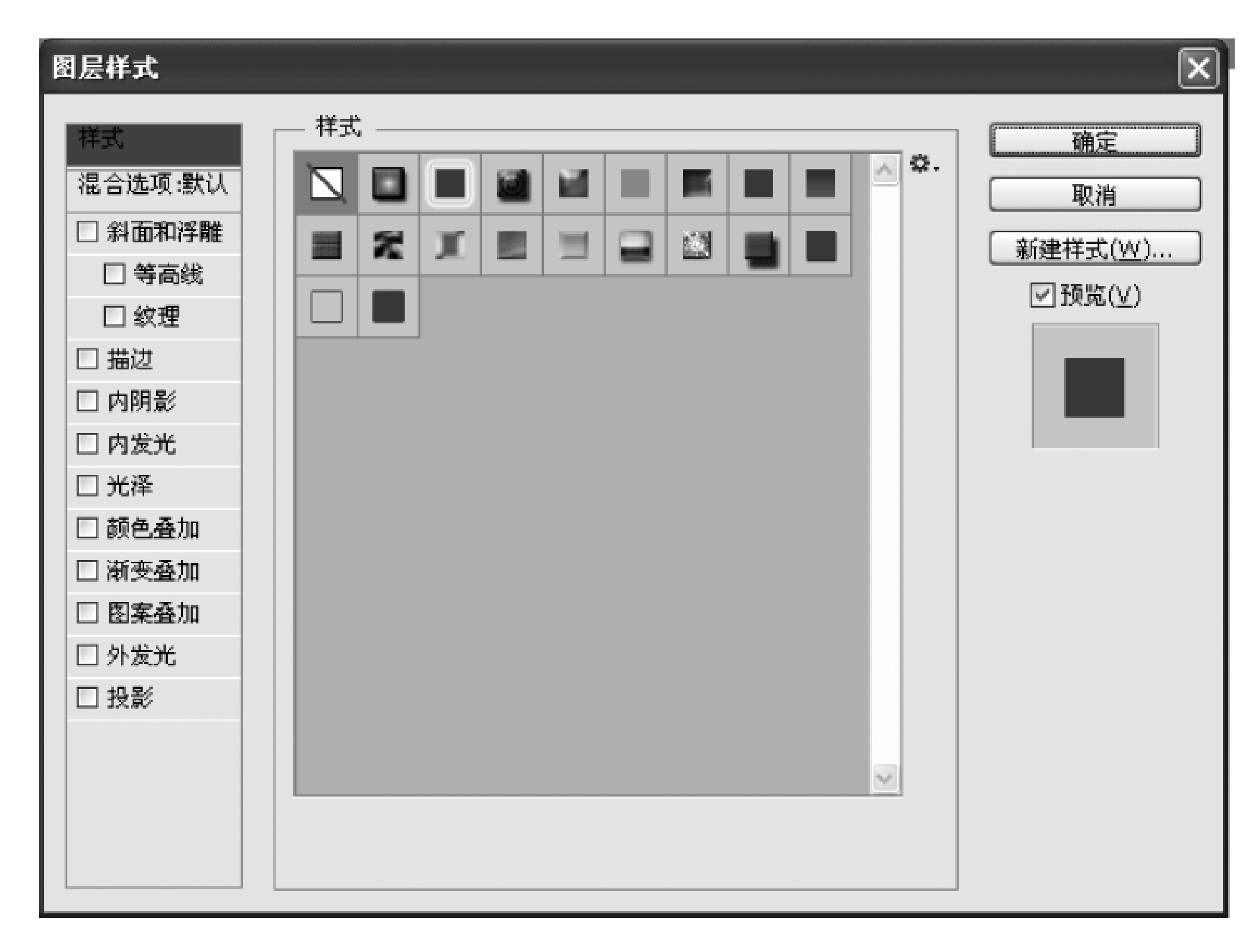


图 4-3　【图层样式】对话框

【混合选项：默认】的图层样式选项包括【投影】、【内阴影】、【外发光】、【内发光】、【斜面和浮雕】、【光泽】、【颜色叠加】、【图案叠加】及【描边】等。勾选不同的选项后，相应的参数设置及选项栏会随之更新，通过设置不同的参数可以产生不同的图层样式效果。

在【图层样式】对话框左侧单击【样式】，右侧更新为 Photoshop 提供的默认样式列表，单

击相应的样式就可以将其应用到当前图层中，单击列表中的【取消样式】按钮可以取消当前图层应用的样式。

2. 保存图层样式

除了可以使用 Photoshop 自带的预设图层样式外，用户还可以将自己设置好的图层样式定义为新的预设样式，在其他文件或以后的工作中调入使用。

选择菜单栏中的【窗口】|【样式】命令，即可打开【样式】面板添加样式。

3. 复制、粘贴和清除图层样式

在 Photoshop 中，图层样式与图像、文字一样，也可以进行复制和粘贴的操作。选中应用样式的图层，然后右击，在快捷菜单中对图层样式进行复制、粘贴和清除操作。除了可以在同一文件的不同图层间进行图层样式的复制、粘贴外，还可以在不同文件的图层间进行相关操作。

4. 图层样式的全局设置

除了可以对图层样式中的各个效果进行单独设置外，还可以对整体效果使用一些命令进行设置，这些命令称为图层样式的全局设置命令。选中应用样式的图层，然后右击，在快捷菜单中选择相应命令进行设置即可。

4.1.4 图层的混合模式

一个图层的混合模式将决定当前图层中的像素与其下面图层中的像素以何种模式进行混合。在【图层】面板左上角有一个【图层混合模式】下拉列表，在此列表中包括 25 种图层混合模式选项，选择不同的选项可以将当前图层设为不同的混合模式，从而使其产生不同的效果。图层混合模式选项如图 4-4 所示。

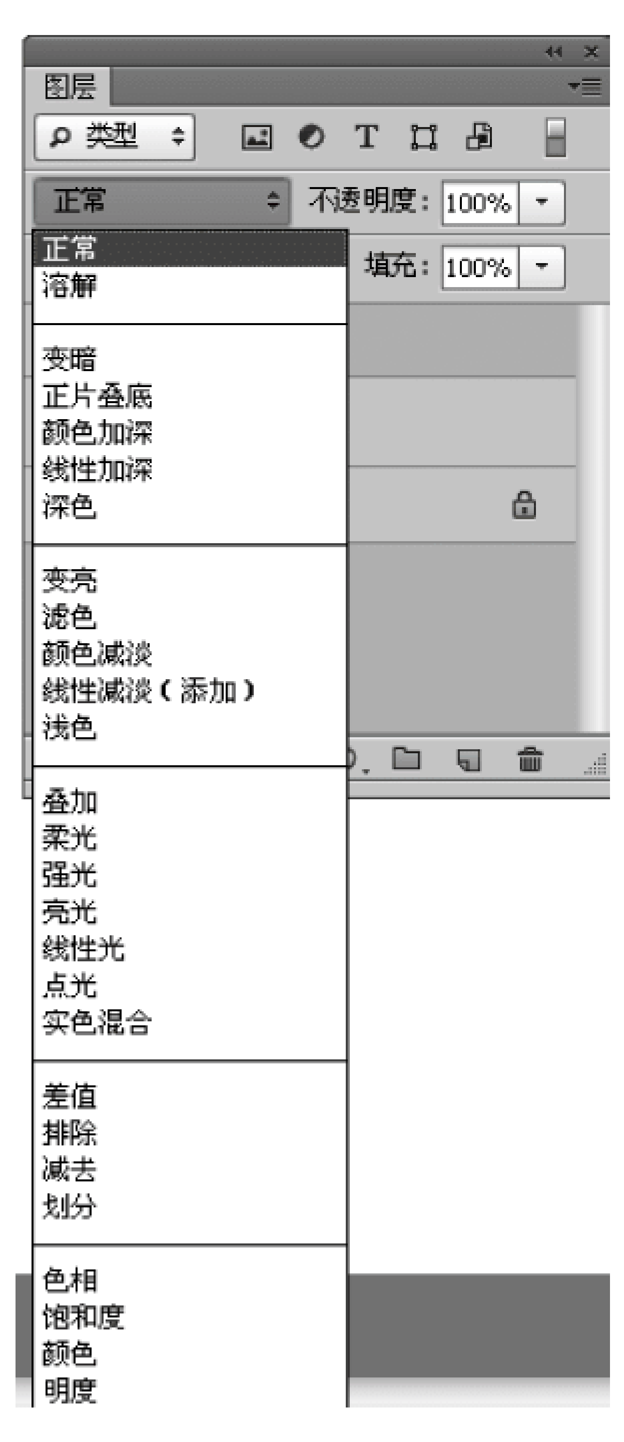


图 4-4 图层混合模式选项

在下面介绍图层模式的时候会反复使用基色、混合色和结果色 3 个术语，为了使读者能够更好地理解这部分内容，首先对这 3 个术语进行介绍。

- 基色：当前图层之下的图层颜色。
- 混合色：当前图层的颜色。
- 结果色：混合后得到的颜色。

1. 两种基本模式

1）【正常】模式

在【正常】模式下编辑或绘制的每个像素都将直接成为结果色，这是默认的模式，也是图像原来的状态。

2）【溶解】模式

使用【溶解】模式，Photoshop 会将当前层中的部分结果色以基色和混合色进行随机替换，替换的程度取决于该像素的不透明度，使用【溶解】模式往往在图像中产生杂点的效果。

2. 5种模式能使图像产生变暗的效果

1)【变暗】模式

使用【变暗】模式，当前图层中的图像会选择基色或混合颜色中较暗的部分作为结果色，比混合色亮的像素将被替换，而比混合色暗的像素不改变，从而使整个图像产生变暗的效果。

2)【正片叠底】模式

【正片叠底】模式根据混合图层图像的颜色将底层图像表现得灰暗。使用【正片叠底】模式的图层查看每个通道中的颜色信息是把基色与混合色相乘，将任何颜色与黑色相乘产生黑色，将任何颜色与白色相乘则颜色保持不变。【正片叠底】模式产生的结果颜色总是相对较暗。

3)【颜色加深】模式

【颜色加深】模式是将两个图层的颜色混合得暗一些，混合后的颜色对比度变强，使得图像整体变得鲜亮。

4)【线性加深】模式

【线性加深】模式通过减小亮度使基色变暗以反映混合色，在混合图层图像的颜色变暗时，背景色也将变暗，在两个颜色混合时，颜色将保持比较清晰的状态，两个图像都能均等的表现。

5)【深色】模式

【深色】模式通过比较混合色和基色的所有通道值的总和并从中选择最小的通道值来创建结果色，从而使整个图像产生变暗的效果。

3. 5种图层模式能使图像产生变亮的效果

1)【变亮】模式

【变亮】模式与【变暗】模式相反，它是通过比较基色与混合色，将比混合色暗的像素替换，而比混合色亮的像素不改变，从而使整个图像产生变亮的效果。

2)【滤色】模式

【滤色】模式与【正片叠底】模式相反，它是查看每个通道的颜色信息，并将混合色的互补色与基色相乘，从而产生相对较亮的效果。

3)【颜色减淡】模式

【颜色减淡】模式与【颜色加深】模式相反，使用【颜色减淡】模式的图层查看每个通道中的颜色信息，使基色变亮以反映混合颜色，与黑色混合则不发生变化。

4)【线性减淡】模式

使用【线性减淡】模式的图层查看每个通道中的颜色信息，并通过增加亮度使基色变亮以反映混合色，与黑色混合则不发生变化。

5)【浅色】模式

【浅色】模式通过比较混合色和基色的所有通道值的总和并从中选择最大的通道值来创建结果颜色，从而使整个图像产生变亮的效果。

4. 不是单纯地将图像变暗或变亮的图层模式

下面 7 种图层模式的计算方法相对复杂，它们不是单纯地将图像变暗或变亮，而是根据图像的具体情况进行不同的处理。

1）【叠加】模式

根据紧临当前图层的下一层图层的颜色使当前图层与其下层图层的颜色产生不同的混合，同时保留下层图层的亮度和暗度，使当前图层产生一种透明的效果。

2）【柔光】模式

【柔光】模式产生的效果与将发散的聚光灯照在图像上相似，它可能使当前颜色变暗，也可能使其变亮。如果当前的混合色比 50%灰色亮，则图像会变亮；如果比 50%灰色暗，则图像会变暗。在使用【柔光】模式的图层中用纯黑色或纯白色作画会产生明显较暗或较亮的区域，但不会产生纯黑色或纯白色的区域。

3）【强光】模式

【强光】模式产生的效果与将耀眼的聚光灯照在图像上相似，它是根据混合色的亮度对当前颜色执行【正片叠底】模式或【屏幕】模式。

4）【亮光】模式

【亮光】模式通过增加或减小图像对比度来加深或减淡颜色。如果混合色比 50%灰色亮，则通过减小对比度使图像变亮；如果混合色比 50%灰色暗，则通过增加对比度使图像变暗。

5）【线性光】模式

【线性光】模式通过减小或增加亮度来加深或减淡颜色，具体取决于混合色。如果混合色比 50%灰色亮，则通过增加亮度使图像变亮；如果混合色比 50%灰色暗，则通过减小亮度使图像变暗。

6）【点光】模式

使用【点光】模式可以根据混合色的不同而产生不同的替换颜色效果。如果混合色比 50%灰色亮，则替换比混合色暗的像素，而不改变比混合色亮的像素；如果混合色比 50%灰色暗，则替换比混合色亮的像素，而不改变比混合色暗的像素。

7）【实色混合】模式

【实色混合】模式取消了中间色的效果，混合的结果只包含纯色。

5. 根据颜色的变化对基色和混合色进行混合产生结果色的模式

1）【差值】模式

【差值】模式是从背景图像的色调中减去混合图层图像的色调来表现两个色调的互补色，混合图层图像的色调越高，效果表现得越强烈。与白色混合会使基色产生反相的效果，与黑色混合不产生变化。

2）【排除】模式

使用【排除】模式可以产生一种与【差值】模式相似但对比度较低的效果。与白色混合会使基色产生反相的效果，与黑色混合不产生变化。

6. 利用基色和混合色的不同属性产生结果色的模式

1)【色相】模式

【色相】模式是用基色的亮度和饱和度以及混合色的色相创建结果色。

2)【饱和度】模式

【饱和度】模式是用基色的亮度和色相以及混合色的饱和度创建结果色,在“0”饱和度(也就是灰色)的区域上使用此模式不会产生变化。

3)【颜色】模式

【颜色】模式是用基色的亮度、混合色的色相和饱和度创建结果色,这样可保留图像中的灰阶,并且对于给单色图像上色和给彩色图像着色非常有用。

4)【明度】模式

【明度】模式是用基色的色相和饱和度以及混合色的亮度创建结果色,【明度】模式产生的效果与【颜色】模式相反。

4.2 通道

通道的主要功能是快速地创建或存储选区,并对复杂图像的选取或制作图像的特殊效果非常有帮助。通道是用来存储图层选取信息的又一特殊图层,在通道中同样可以进行绘图与编辑等操作。

4.2.1 通道的基本概念

在 Photoshop 中,通道主要用来保存图像的色彩信息和选区,可分为两大类,即颜色通道和 Alpha 通道。颜色通道主要用来保存图像的色彩信息,Alpha 通道主要用来保存选区,但这并不是一成不变的。

1. 颜色通道

保存图像颜色信息的通道称为颜色通道。每个图像都有一个或多个颜色通道,图像中默认的颜色通道数取决于其颜色模式。例如,CMYK 图像默认有 4 个通道,分别代表青色、洋红、黄色和黑色信息。在默认情况下,位图模式、灰度、双色调和索引颜色图像只有一个通道,RGB 和 Lab 图像有 3 个通道,CMYK 图像有 4 个通道。

每个颜色通道都存放着图像中颜色元素的信息,所有颜色通道中的颜色叠加混合将产生图像中像素的颜色。读者可以将通道看作印刷中的印版,即单个印版对应每个颜色图层。

在【通道】面板中通道都显示为灰色,它通过不同的灰度表示 0~255 级亮度的颜色。因为通道的效果较难控制,通常不直接修改颜色通道改变图像的颜色。

除了默认的颜色通道外,用户还可以在图像中创建专色通道,例如在图像中添加黄色、紫色等通道。在图像中添加专色通道后,必须将图像转换为多通道模式。

2. Alpha 通道

除了颜色通道外,用户还可以在图像中创建 Alpha 通道,以便于保存和编辑蒙版及选

区。用户可以在【通道】面板中创建 Alpha 通道，并根据需要进行编辑，然后再调用选区；也可以在图像中建立选区，然后选择【选择】|【存储选区】命令，将现有的选区保存为新的 Alpha 通道。

Alpha 通道也使用灰度表示，其中白色部分对应 100%选择的图像，黑色部分对应未选择的图像，灰色部分表示相应的过渡选择，即选区有相应的透明度。

Alpha 通道也可以转换为颜色通道。

4.2.2 【通道】面板

1. 从通道载入选区

按住 Ctrl 键，在【通道】面板上单击通道的缩略图，可以根据该通道在图像窗口中建立新的选区。

如果图像窗口中已有选区，可进行以下操作：

- 按住 Ctrl＋Alt 键，在【通道】面板上单击通道的缩略图，新生成的选区是从原选区减去根据该通道建立的选区部分。
- 按住 Ctrl＋Shift 键，在【通道】面板上单击通道的缩略图，根据该通道建立的选区添加至原选区。
- 按住 Ctrl＋Shif＋Alt 键，在【通道】面板上单击通道的缩略图，根据该通道建立的选区与原选区重叠的部分作为新的选区。

2.【通道】面板简介

在 Photoshop 提供的【通道】面板中可以创建和管理通道，并观察编辑效果。在【通道】面板上列出了当前图像中的所有通道，最上方是复合通道(在 RGB、CMYK 和 Lab 图像中，复合通道为各个颜色通道叠加的效果)，然后是单个颜色通道、专色通道，最后是 Alpha 通道，如图 4-5 所示。

1）功能按钮

【通道】面板下方有 4 个功能按钮，将鼠标指针移至按钮图标处会出现按钮的功能介绍，单击相应的按钮完成相应的设置。

2）通道的基本操作

通道的创建、复制等操作与图层相似，这里不再详细介绍。

在【通道】面板中单击复合通道会同时选择复合通道及颜色通道，此时在图像窗口中将显示图像的效果，可以对图像进行编辑。

单击除复合通道外的任意通道，在图像窗口中将显示出相应的通道效果，此时可以对选择的通道进行编辑。

按住 Shift 键可以同时选择几个通道，在图像窗口中显示被选择通道的叠加效果。



图 4-5 【通道】面板

单击通道左侧的按钮可以隐藏其对应的通道效果，再次单击该按钮，则可以将通道的效果显示出来。

3）【通道】面板菜单

单击【通道】面板右上角的按钮，将弹出【通道】面板菜单，如图 4-6 所示。

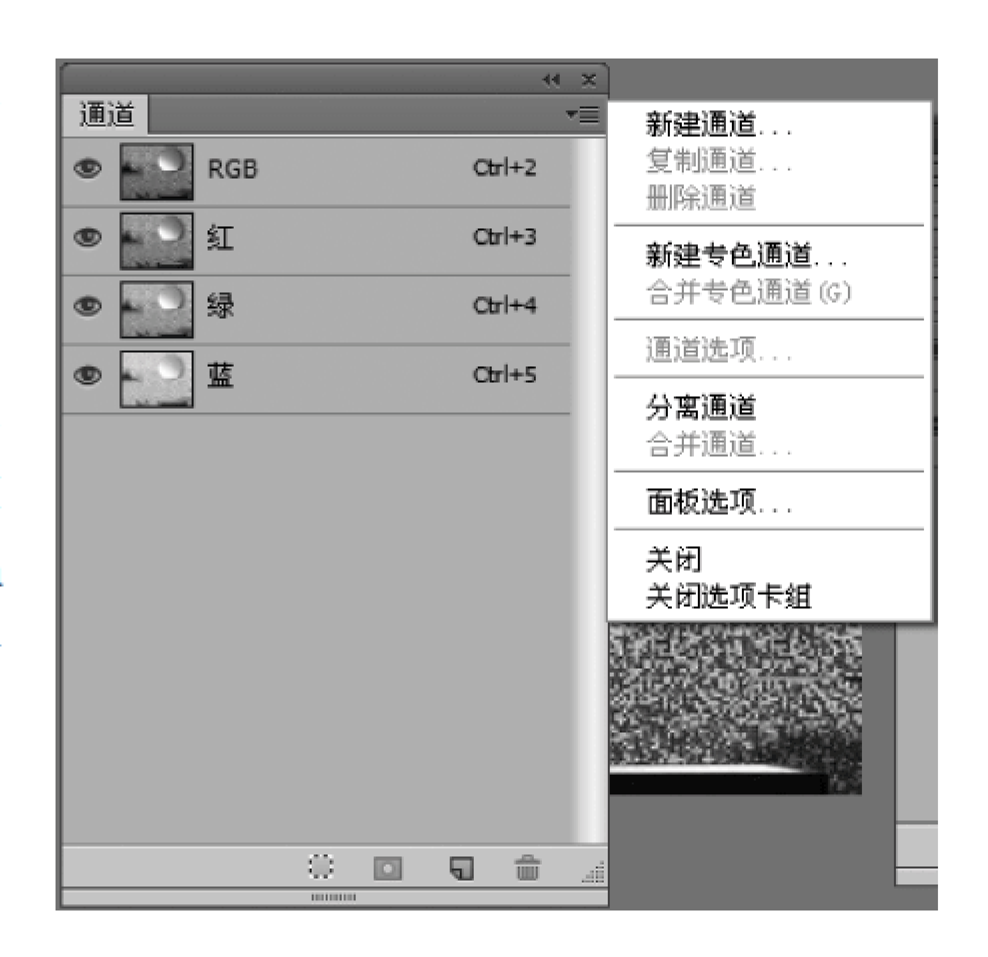


图 4-6 【通道】面板菜单

4）创建新 Alpha 通道

【新建通道】命令用于创建新的 Alpha 通道，选择该命令，在【名称】文本框中输入新 Alpha 通道的名称即可。在【色彩指示】选项栏中选择【被蒙版区域】单选按钮，将创建一个黑色的 Alpha 通道；选择【所选区域】单选按钮，则将创建一个白色的 Alpha 通道。

5）复制通道

【复制通道】命令用来对当前通道进行复制，使用该命令复制出的新通道是 Alpha 通道。在【复制通道】对话框中可以设置名称、位置等选项，若勾选【反相】复选框，新复制的通道是当前通道的反相效果，也就是说它的黑和白完全相反。

6）删除通道

【删除通道】命令用于删除多余的通道，在【通道】面板中选择要删除的通道并选择该命令，可以将当前被选择的通道删除。

7）新建专色通道

使用【新建专色通道】命令可以创建一个新的颜色通道，这种颜色通道只能在图像中产生一种颜色，所以也称专色通道。选择【新建专色通道】命令，在弹出的【新建专色通道】对话框中对颜色、明度进行设置。

8）合并专色通道

【合并专色通道】命令只有在图像中创建了新的专色通道后才可用，图像中颜色通道的数量和类型是受图像的颜色模式控制的，使用该命令可以将新的专色通道合并到图像默认的颜色通道中。

9）设置通道选项

【通道选项】命令只有在选择了 Alpha 通道和新创建的专色通道时才起作用，它主要用来设置 Alpha 通道和专色通道的选项。

10）【分离通道】

选择【分离通道】命令可以将图像中的每一个通道分离为一个单独的灰度图像。

11）合并通道

使用【合并通道】命令必须满足 3 个条件，一是作为通道进行合并的图像的颜色模式必须是灰度的；二是这些图像的长度、宽度和分辨率必须完全相同；三是它们必须是已经打开的。选择【合并通道】命令，在【合并通道】对话框中对模式、通道等进行设置，单击【确定】按钮完成此操作。

4.3 蒙版

在实际的设计工作中，对图像的调整、删除等操作若直接作用于图像，在经过多次操作后如果对前期某步的调整不满意，再来重新调整会很费事，对于这种情况通常使用蒙版功能避免，利用蒙版可以在不破坏图像的前提下对图像进行灵活编辑。

4.3.1 蒙版的基本操作

在图像中，图层蒙版的作用是根据蒙版中颜色的变化使其所在层图像的相应位置产生透明效果。在图层蒙版中使用灰度颜色，其中，当前图层中与蒙版的白色部分相对应的图像不产生透明效果，与蒙版的黑色部分相对应的图像完全透明，与蒙版的灰色部分相对应的图像根据其灰度产生相同程度的透明效果。

(1) 单击图层缩略图，当缩略图以白边显示时表示当前编辑的是图像内容。

(2) 单击图层蒙版缩略图，当缩略图以白边显示时表示当前编辑的是蒙版内容。

1. 创建图层蒙版

在图像中建立选区，单击【图层】面板中的【添加图层蒙版】按钮，即在当前图层上创建图层蒙版，默认添加的图层蒙版显示选区内的图像。

选中图层，单击【图层】面板中的【添加图层蒙版】按钮，即添加白色蒙版。

2. 停用/启用图层蒙版

在【图层】面板的图层蒙版缩览图上右击，在弹出的快捷菜单中选择【停用图层蒙版】命令，图层蒙版即被停用，失去作用，此时图层蒙版缩览图上将出现一个红色的“×”号。

3. 删除图层蒙版

在【图层】面板的图层蒙版缩览图上右击，在弹出的快捷菜单中选择【删除图层蒙版】命令，此时图层蒙版将被删除，原图像不受影响。

4. 应用图层蒙版

在【图层】面板的图层蒙版缩览图上右击，在弹出的快捷菜单中选择【应用图层蒙版】命令，此时图层蒙版将被删除，原图像只保留使用图层蒙版时可见的部分，其他部分也被删除。

4.3.2 快速蒙版

在工具箱下方有两个按钮，即【以标准模式编辑】按钮和【以快速蒙版模式编辑】按钮。单击【以标准模式编辑】按钮，使用正常的编辑模式，此时所进行的操作对当前图像起作用；单击【以快速蒙版模式编辑】按钮，以快速蒙版的方式进行编辑，此时所进行的操作对图像本身不产生作用，而是对当前在图像中产生的快速蒙版进行编辑。

1. 快速蒙版与图层蒙版的区别

使用图层蒙版需要在【通道】面板中保存蒙版，它的主要功能是根据蒙版中颜色的变化使其所在层图像的相应位置产生透明效果。

在使用快速蒙版时，【通道】面板中会出现一个临时的快速蒙版通道，但操作结束后不在【通道】面板中保存该蒙版，而是直接生成选区。快速蒙版常被用来创建各种特殊选区，从而制作出特别的图像效果。

2. 使用快速蒙版创建选区

使用快速蒙版创建选区有以下几个步骤：

（1）打开或新建一幅图像。

（2）单击工具箱下方的【以快速蒙版模式编辑】按钮，进入快速蒙版编辑状态。

（3）利用工具箱中的工具编辑快速蒙版。

（4）单击工具箱中的【以标准模式编辑】按钮，回到标准编辑模式，此时图像上出现利用快速蒙版创建的选区，下面进行的编辑操作将重新对图像起作用。

3. 设置快速蒙版选项

双击工具箱中的【以快速蒙版模式编辑】按钮，弹出【快速蒙版选项】对话框。

选择【被蒙版区域】单选按钮，快速蒙版中不显示色彩的部分将作为最终的选区；选择【所选区域】单选按钮，快速蒙版中显示色彩的部分将作为最终的选区。

【颜色】色块决定了快速蒙版在图像窗口中显示的色彩；【不透明度】值决定了快速蒙版的最大不透明效果值。

4.4　综合应用

4.4.1　按钮的制作

按钮的制作步骤如下：

（1）选择菜单栏中的【文件】|【新建】命令，新建一个500像素×400像素的文件，背景填充颜色#033706，效果如图4-7所示。

（2）新建一个图层，选择矩形选框工具，拖出一个288像素×50像素的矩形选区，填充颜色#036208，效果如图4-8所示。



图4-7　新建文件



图4-8　建立矩形选区

（3）选中矩形所在的图层，单击【图层】面板中的【图层样式】按钮，设置内阴影和渐变叠加，参数分别如图 4-9 和图 4-10 所示。



图 4-9　内阴影参数设置



图 4-10　渐变叠加参数设置

(4) 选中第 1 个矩形，按 Ctrl＋Alt＋T 键，移动第 2 个矩形至目标位置，按 Enter 键确认。然后按 Ctrl＋Alt＋Shift＋T 键，等距离复制出两个按钮，效果如图 4-11 所示。

(5) 选择工具箱中的横排文字工具，在每个矩形上输入菜单文字，字体为“Arial”，效果如图 4-12 所示。

(6) 单击【图层】面板下方的【新建图层】按钮，新建一个图层，然后选择椭圆选框工具，在矩形的左边拉出一个直径为 30 像素的正圆选区，并填充颜色为＃033706，效果如图 4-13 所示。

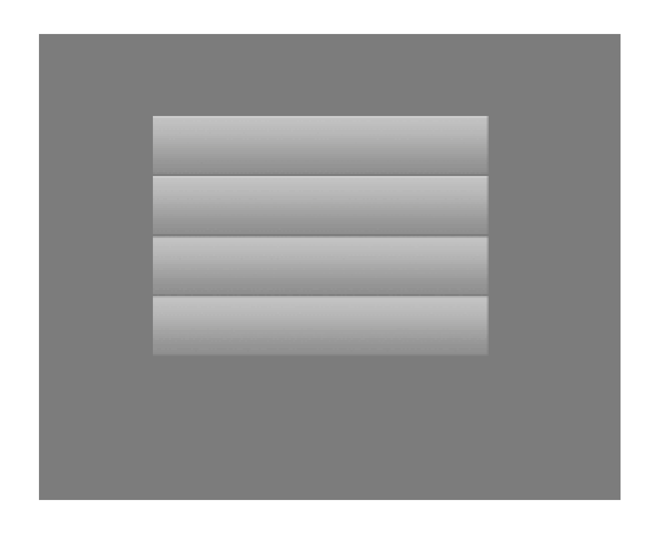




图 4-11 复制矩形后的效果

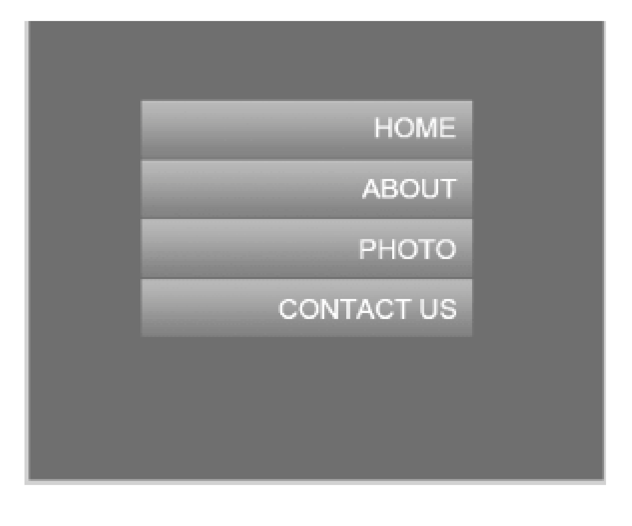


图 4-12 添加文字效果

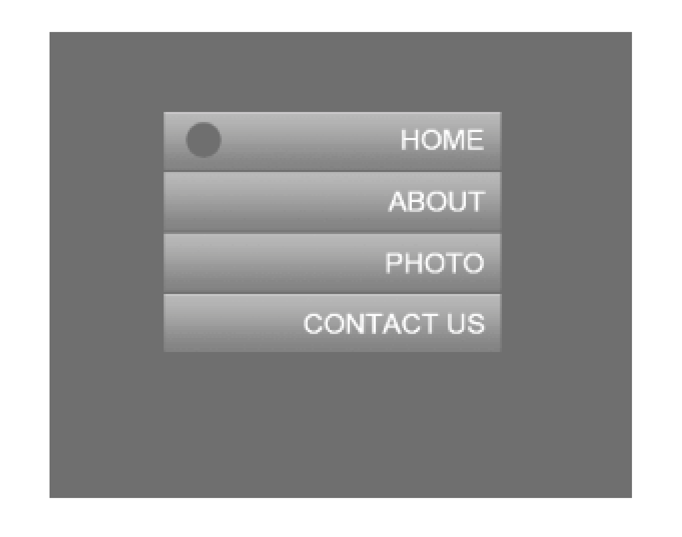


图 4-13 添加正圆选区

(7) 为绿色正圆加上图层样式，首先选中绿色正圆所在的图层，单击【图层】面板下方的【图层样式】按钮，分别设置内发光、渐变叠加、描边，参数设置如图 4-14～图 4-16 所示。

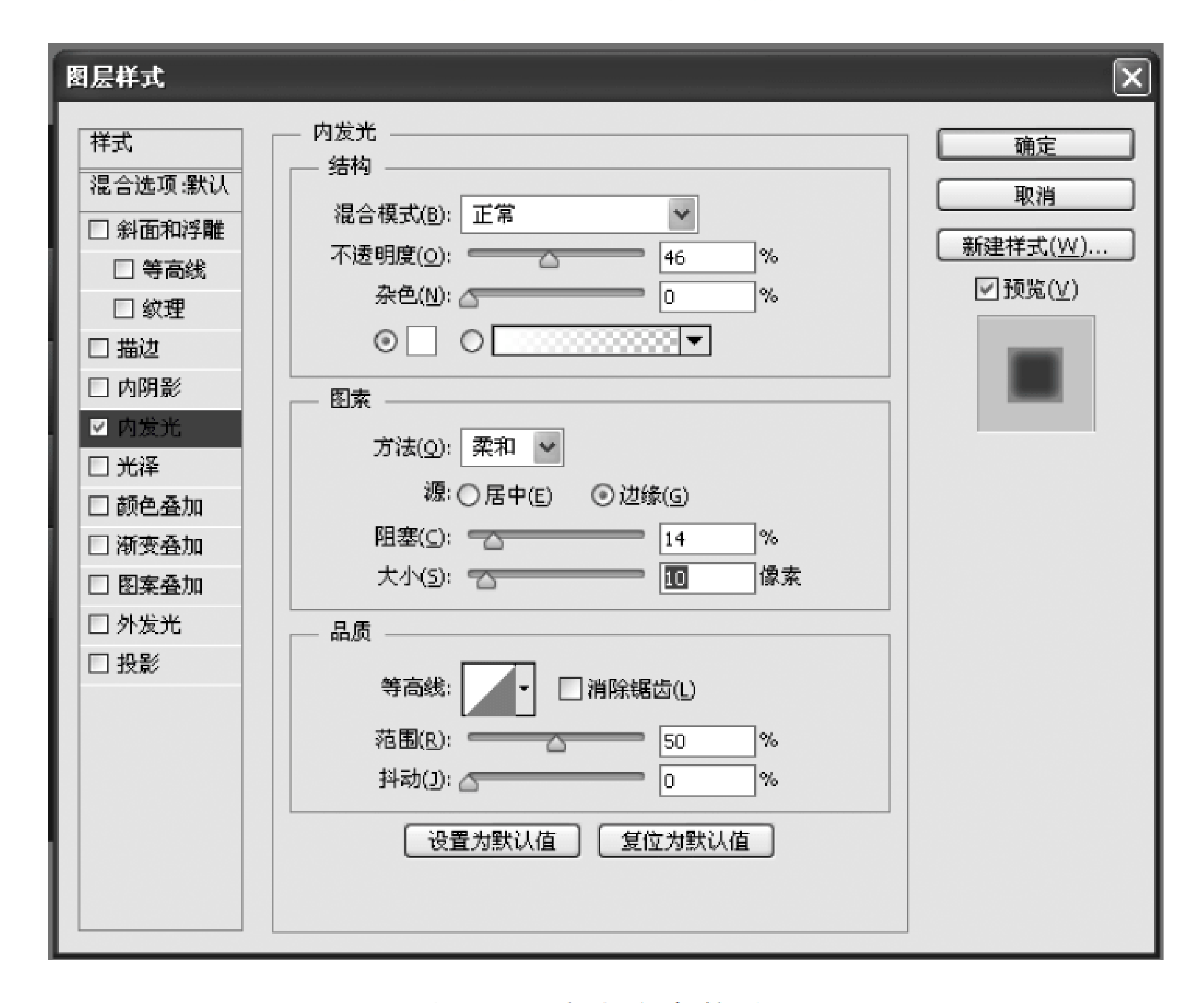


图 4-14 内发光参数设置

(8) 加上图层样式后把正圆的图层不透明度改为 70%，然后在上面加上箭头，效果如图 4-17 所示。

(9) 稍微加一些背景颜色，完成最终效果，如图 4-18 所示。



图 4-15 渐变叠加参数设置

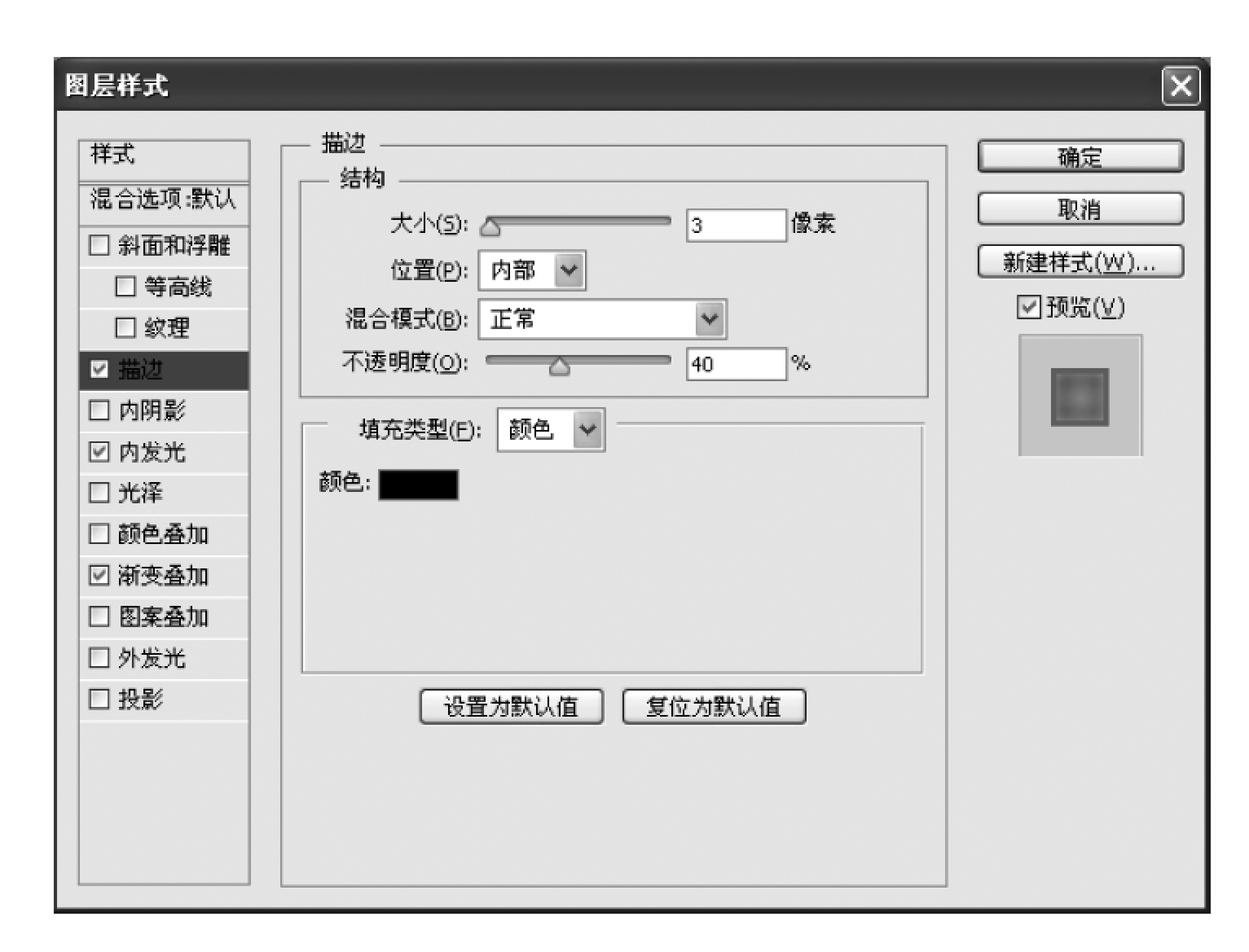


图 4-16 描边参数设置

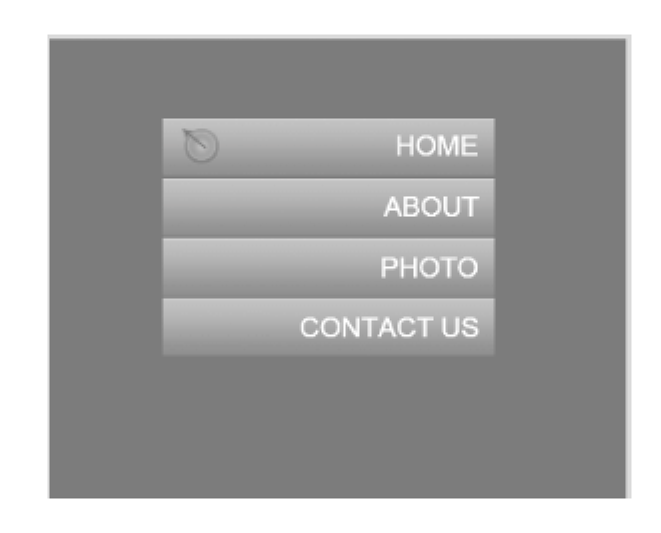


图 4-17 修改正圆选区

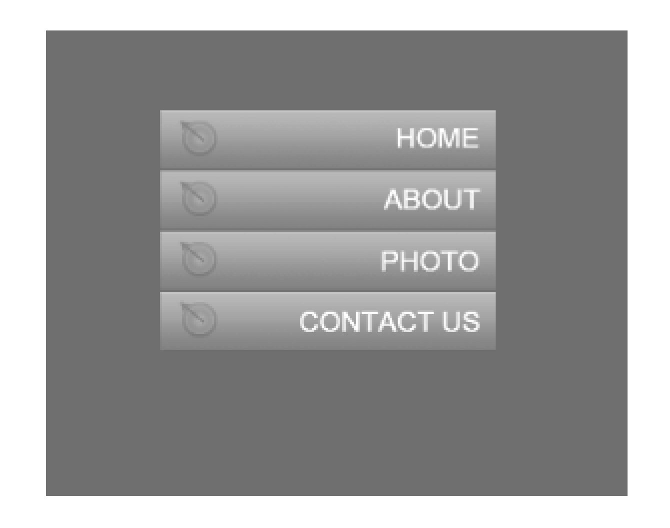


图 4-18 最终效果

4.4.2　蒙版的合成

蒙版合成的操作步骤如下：

(1) 选择菜单栏中的【文件】|【打开】命令，打开“素材 1”和“素材 2”两张素材图片，如图 4-19 所示。

(2) 将“素材 2”拖至“素材 1”图中，【图层】面板如图 4-20 所示。



图 4-19　打开素材图片



图 4-20　【图层】面板

(3) 单击图层缩览图前面的眼睛图标，隐藏图层 1；在【通道】面板中的任一单色通道内按住 Ctrl 键单击通道缩略图，将选区图像调入选区，如图 4-21 所示。

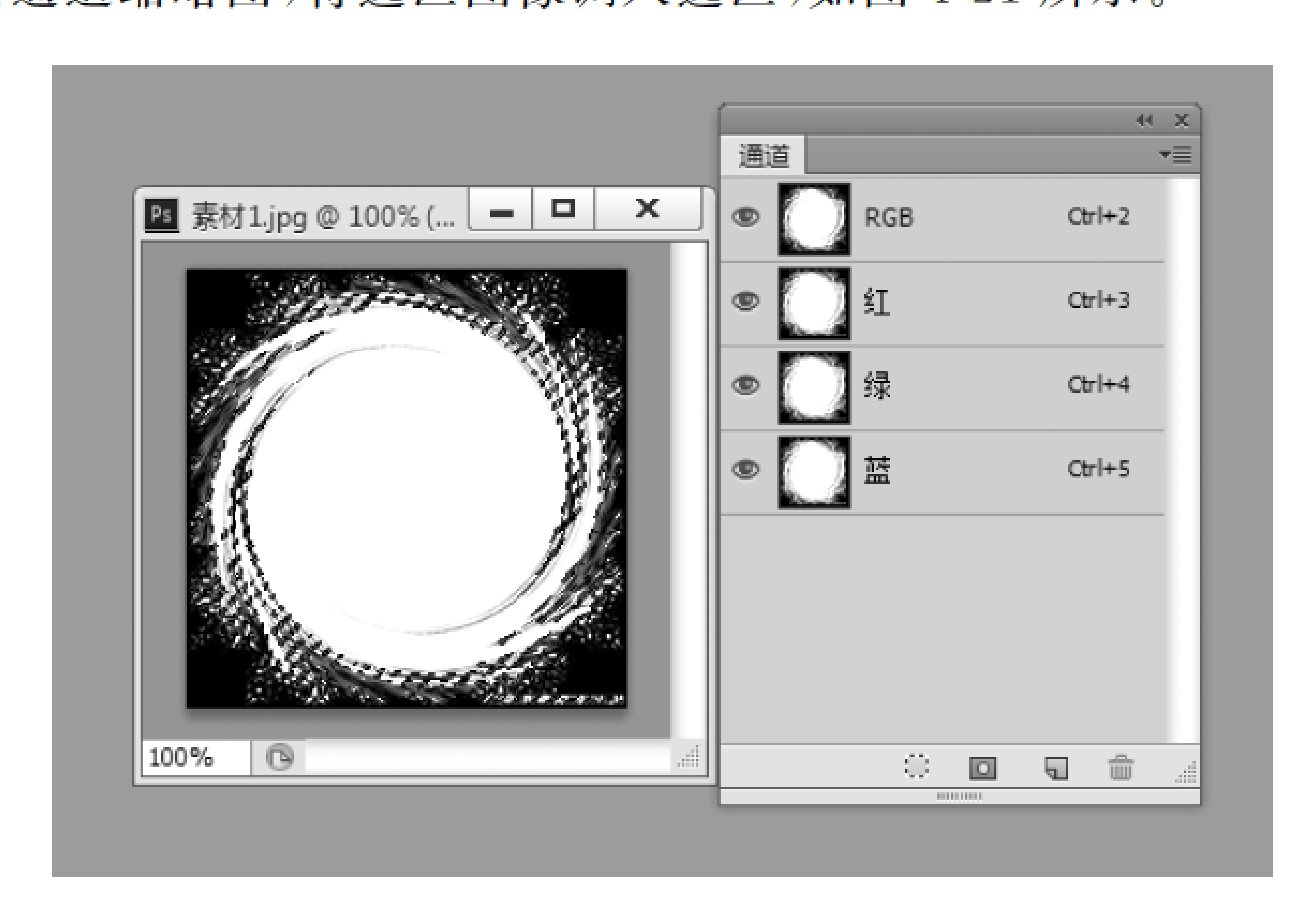


图 4-21　将选区图像调入选区

(4) 返回【图层】面板，再次单击图层缩览图前面的眼睛图标，显示图层 1；选中图层 1，单击【图层】面板下方的【添加蒙版】按钮，为图层 1 添加蒙版，如图 4-22 所示。

(5) 单击【图层】面板下方的【创建新图层】按钮，建立新图层，将【前/背景色】调为蓝/白色，在渐变填充工具中选择径向渐变，在图层 2 中拖动，添加渐变效果，如图 4-23 所示。



图 4-22　为图层 1 添加蒙版

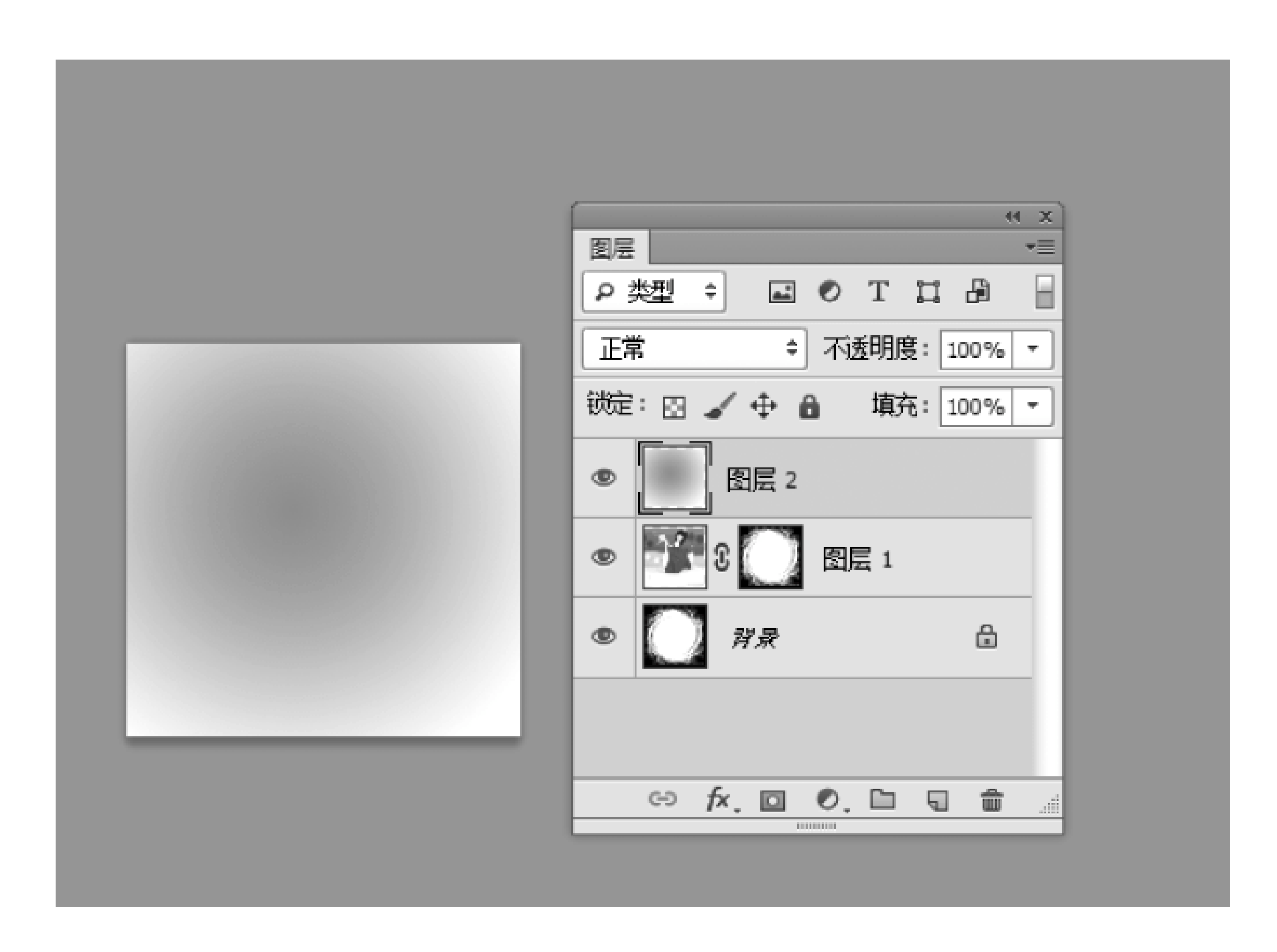


图 4-23　建立新图层

(6) 将图层 2 移动至图层 1 下方，然后保存，最终效果如图 4-24 所示。



图 4-24　最终效果

4.4.3 艺术相框的制作

制作艺术相框的操作步骤如下：

(1) 选择菜单栏中的【文件】|【新建】命令，建立一个图形文件，参数设置如图 4-25 所示。



图 4-25 新建文件的参数设置

(2) 打开“素材 1”图片，将其拖入刚才建立的文件中，并调整大小及位置，如图 4-26 所示。

(3) 选择多边形套索工具，绘制特殊形状的选区，然后在【通道】面板下方单击【将选区存储为通道】按钮，把选区保存起来，并取消选区，如图 4-27 所示。

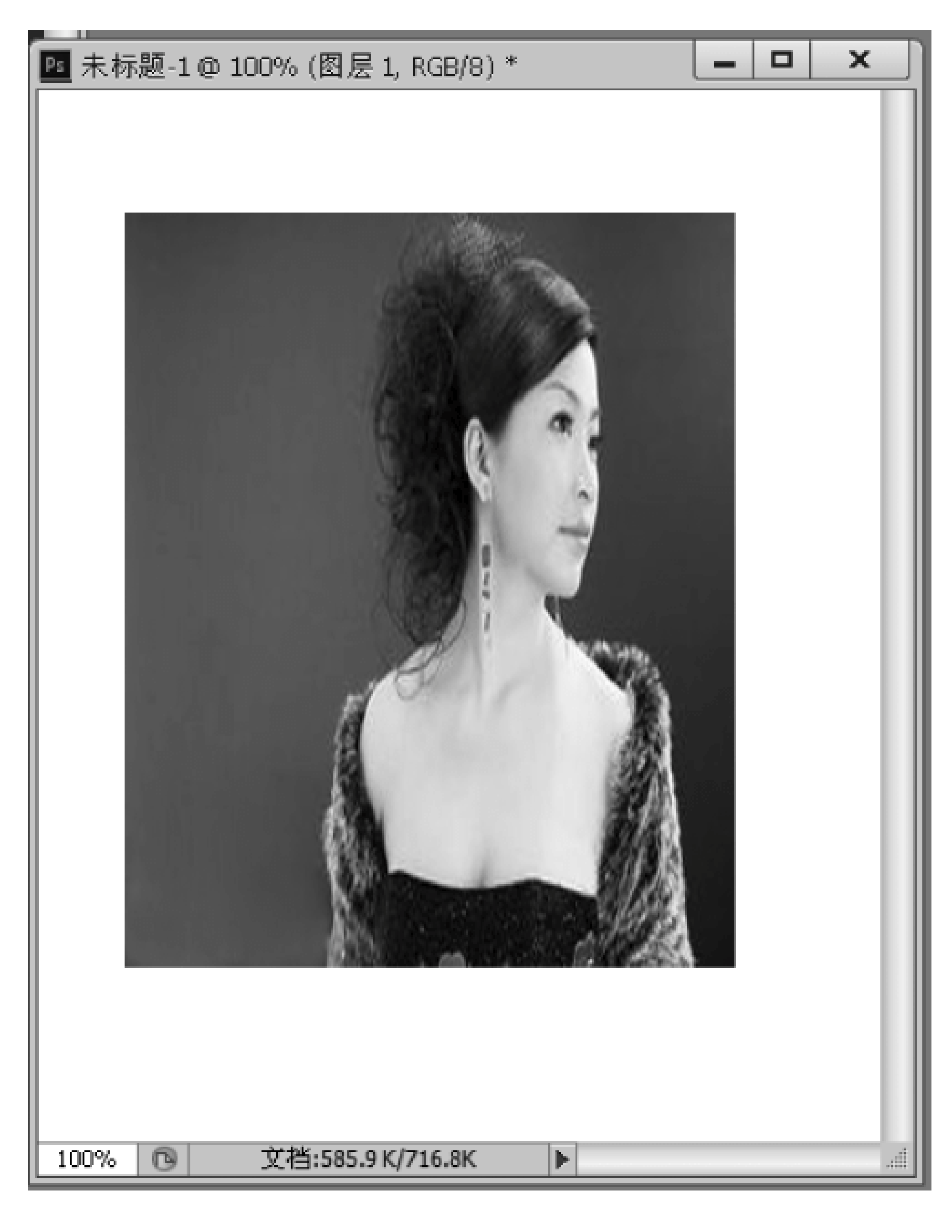


图 4-26 打开素材



图 4-27 制作选区

(4) 选择菜单栏中的【滤镜】|【滤镜库】|【画笔描边】|【喷溅】命令，打开【喷溅】对话框，设置相应参数，如图 4-28 所示。

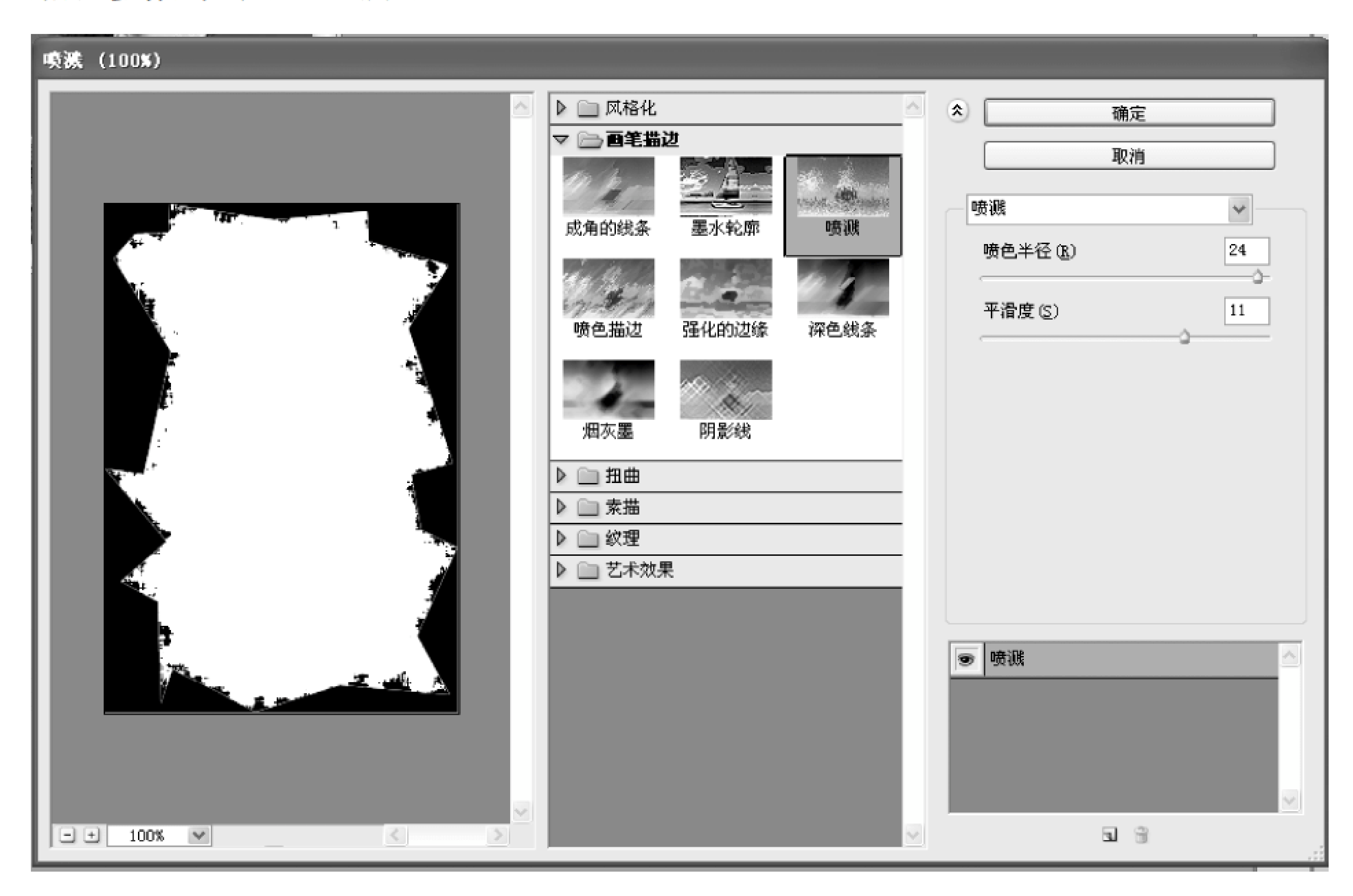


图 4-28 喷溅参数设置

(5) 在【通道】面板下方单击【将通道作为选区载入】按钮，得到不规则选区，然后将选区反选，并按 Delete 键删除选中部分的图像，如图 4-29 所示。



图 4-29 将选区反选

（6）为图层 1 添加投影样式，如图 4-30 所示。

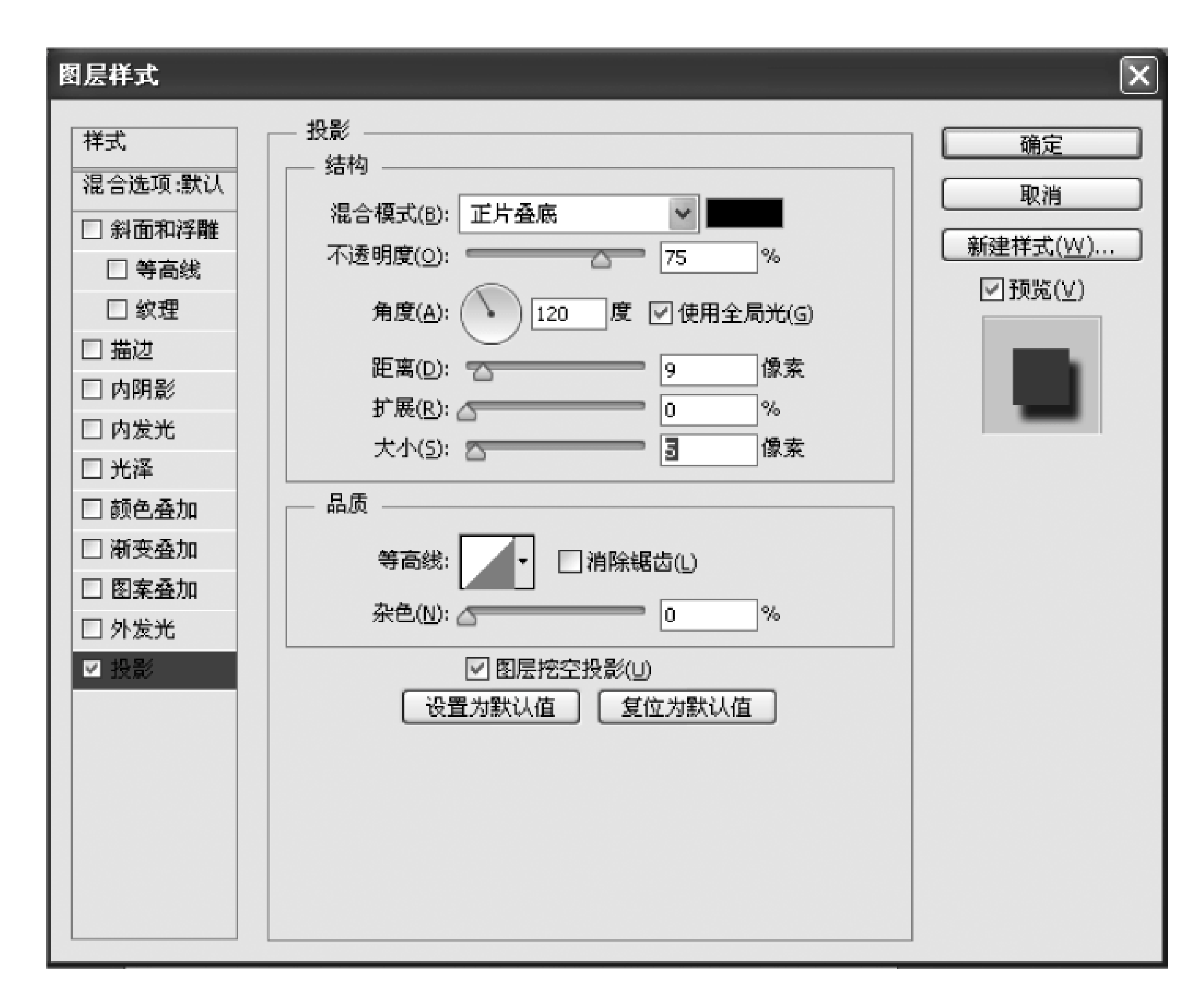


图 4-30 投影参数设置

（7）选择菜单栏中的【文件】|【保存】命令保存文件，最终效果如图 4-31 所示。



图 4-31 最终效果

第5章 Photoshop色彩调整及滤镜

本章学习目标：

✍了解图像的色彩相关知识。

✍掌握图像的编辑操作。

✍掌握图像的色彩调整常用命令。

✍掌握 Photoshop 中滤镜的应用。

5.1 图像的编辑

【编辑】菜单中的命令用于对图像文件进行纠正、修改及剪贴等处理，主要包括还原、复制、粘贴、清除、填充、描边、自由变换、变换、定义图案及清理等命令，【编辑】菜单如图 5-1 所示。



图 5-1 【编辑】菜单

1. 还原操作

【还原】命令主要用于对图像编辑处理过程中出现的失误进行复原。

在【编辑】菜单中，这一命令的名称经常显示为“还原＋上一步操作名称”，此命令的主要功能是将图像文件恢复到最后一次操作前的状态，执行该命令后，它变成“重做＋上一步操作名称”命令，其快捷键为 Ctrl＋Z。

选择【前进一步】命令，当图像中有被撤销的操作时，向前重做一步操作，其快捷键为 Shift＋Ctrl＋Z。

选择【后退一步】命令，向后撤销一步操作，其快捷键为 Alt＋Ctrl＋Z。

2. 图像的复制

图像的复制与粘贴命令主要包括【剪切】、【拷贝】、【合并拷贝】、【粘贴】等，这些命令在实际工作中的使用非常频繁，而且经常配合使用，其中【剪切】、【拷贝】和【粘贴】命令的功能比较明确，这里不做介绍，下面只简单介绍【合并拷贝】和【贴入】

命令的功能。

【合并拷贝】命令主要用于图层文件，用于将所有图层中的内容复制到剪贴板中，在进行粘贴时，将其合并到一个图层粘贴。

【贴入】命令在【选择性粘贴】命令下，在使用该命令时，当前图像文件中必须有选区，可将剪贴板中的内容粘贴到当前图像文件中，并将选区设置为图层蒙版。

3. 图像的填充与描边

使用【编辑】|【填充】命令可以将选定的内容按指定的模式填入图像的选区内或直接将其填入图层内，使用【编辑】|【描边】命令可以用前景色沿选区边缘描绘指定宽度的线条。

4. 图像的变换

图像的变换命令在实际工作过程中经常运用，用户熟练地掌握此类命令，可以绘制较强的图像效果。

- 【自由变换】命令：在自由变换状态下，以手动方式将当前图层的图像或选区做任意缩放、旋转等自由变形操作。这一命令用在路径上时会变为【自由变换路径】命令，对路径进行自由变换。
- 【变换】命令：主要包括【缩放】、【旋转】、【斜切】、【扭曲】、【透视】、【变形】、【旋转 180 度】、【旋转 90 度(逆时针)】、【旋转 90 度(顺时针)】、【水平翻转】及【垂直翻转】等命令，用户可以根据不同的需要选择不同的命令，对图像或选区进行变换调整。这一命令用在路径上时会变为【变换路径】命令，用于对路径进行变换。

5. 调整图像大小

在 Photoshop 中可以利用【图像】|【图像大小】命令重新设定图像文件的尺寸大小和分辨率。

选择要调整的图像，然后选择【图像】|【图像大小】命令，打开【图像大小】对话框进行设置。

- 【像素大小】类参数和【文档大小】类参数主要用于设置修改后图像的大小，这两组参数只要修改其中的一组，另一组就会随之发生相应的变化。
- 如果图像带有应用了样式的效果层，那么勾选【缩放样式】复选框可以在缩放图像的同时缩放样式效果。【缩放样式】选项只有在勾选了【约束比例】复选框时才能使用。
- 勾选【约束比例】复选框，可以对图像进行等比例缩放。
- 勾选【重定图像像素】复选框，可以在其下拉列表中选择修改图像大小时使用的插值方法。

6. 调整画布大小

利用【图像】|【画布大小】命令可以重新设定图像版面的尺寸大小，并可以调整图像在版面上的位置。

7. 旋转画布

利用【图像】|【旋转画布】命令可以调整图像版面的角度，并且文件中的所有图层、通道以及路径会一起旋转。

5.2 颜色的调整

选择【图像】|【调整】命令，系统将弹出如图 5-2 所示的级联菜单，该命令比较重要，主要用于调整图像的色调、亮度、对比度以及饱和度等，利用它能够调出漂亮的色彩画面效果。



图 5-2 【调整】菜单

1.【色阶】命令

【色阶】命令主要用来调整各通道的明暗数量。选择【色阶】命令，在弹出的【色阶】对话框中对参数进行设置，如图 5-3 所示。

- 【通道】：在此下拉列表中选择要调整的通道。
- 【输入色阶】值和【色阶】对话框中间的【色阶】直方图是相对应的，调整【输入色阶】值或移动滑块的位置可以修改图像中的明暗数量及图像对比度。
- 【色阶】直方图中最左侧的黑色滑块代表当前图像的最暗值，移动该滑块时，其左侧的亮度级全部修改为原图像中的最暗值；【色阶】直方图最右侧的白色滑块代表当前图像中的最亮值，移动该滑块时，其右侧的亮度级全部修改为原图像中的最亮值；【色阶】直方图中左侧滑块至右侧滑块的间距就是颜色由最暗过渡到最亮的过程，中间的灰色滑块表示当前图像的中间亮度值；【色阶】直方图中的高度表示图像中每个亮度级的数量。直方图下方有 3 个滑块，当滑块向左滑动时增加图像中暗调的数量，向右滑动时增加图像中亮调的数量，3 个滑块间的距离越小，图像的对比度越大。



图 5-3 【色阶】对话框

- 【输出色阶】值和【色阶】对话框最下方的【色阶】色带是相对应的，调整【输出色阶】值和【色阶】色带的滑块可以调整整个图像的亮度和对比度。【色阶】色带左侧的黑色滑块表示图像的最暗值，右侧的无色滑块表示图像中的最亮值。滑块间的距离越小，图像的对比度越小。

2.【自然饱和度】命令

【自然饱和度】命令的功能和【色相/饱和度】命令类似，可以使图片更加鲜艳或暗淡，但效果更加细腻，会智能地处理图像中不够饱和的部分和忽略足够饱和的颜色。在使用【自然饱和度】调整图像时会自动保护图像中已饱和的部位，只对其做小部分的调整，而着重调整不饱和的部位，这样会使 Photoshop CS6 图像整体的饱和趋于正常。【自然饱和度】对话框如图 5-4 所示。



图 5-4 【自然饱和度】对话框

- 【自然饱和度】：主要针对图像中饱和度过低的区域增加饱和度。
- 【饱和度】：对色彩的饱和度起主要作用。

3.【颜色查找】命令

【颜色查找】命令是 Photoshop CS6 中文版新增的功能，主要作用是对 Photoshop CS6 图像色彩进行校正，校正的方法有【3DLUT 文件】(三维颜色查找表文件，精确校正图像色彩)、【摘要】、【设备链接】，并且可以打造一些特殊图像效果，其对话框如图 5-5 所示。

- 【3DLUT 文件】：右侧的预设下拉列表中显示预设效果。
- 【摘要】：选择该单选按钮时会弹出【载入】对话框，用来载入需要载入的色彩文件或者在【摘要】右侧的预设下拉列表中选择一种预设效果。
- 【设备链接】：在右侧的预设下拉列表中选择相应的预设效果。



图 5-5　颜色查找对话框

4.【HDR 色调】命令

Photoshop CS6 新增的【HDR 色调】命令可用来修补太亮或太暗的图像，制作出高动态范围的 Photoshop CS6 图像效果，其对话框如图 5-6 所示。



图 5-6　【HDR 色调】对话框

- 【预设】：系统默认设置的预设值。
- 【方法】：选择系统中设置的方法。
- 【半径】：控制发光效果的大小。
- 【强度】：控制发光效果的对比度。
- 【平滑边缘】：提升细节时启用边缘保留平滑。
- 【灰度系数】：调整高光和阴影之间的差异。

- 【曝光度】：调整图像的整体色调。
- 【细节】：查找图像细节。
- 【阴影】：调整阴影区域的明亮度。
- 【高光】：调整高光区域的明亮度。

5.【曲线】命令

使用【曲线】命令可以利用曲线调整图像各通道的明暗数量。选择【图像】|【调整】|【曲线】命令，在弹出的【曲线】对话框中进行相应设置，如图 5-7 所示。

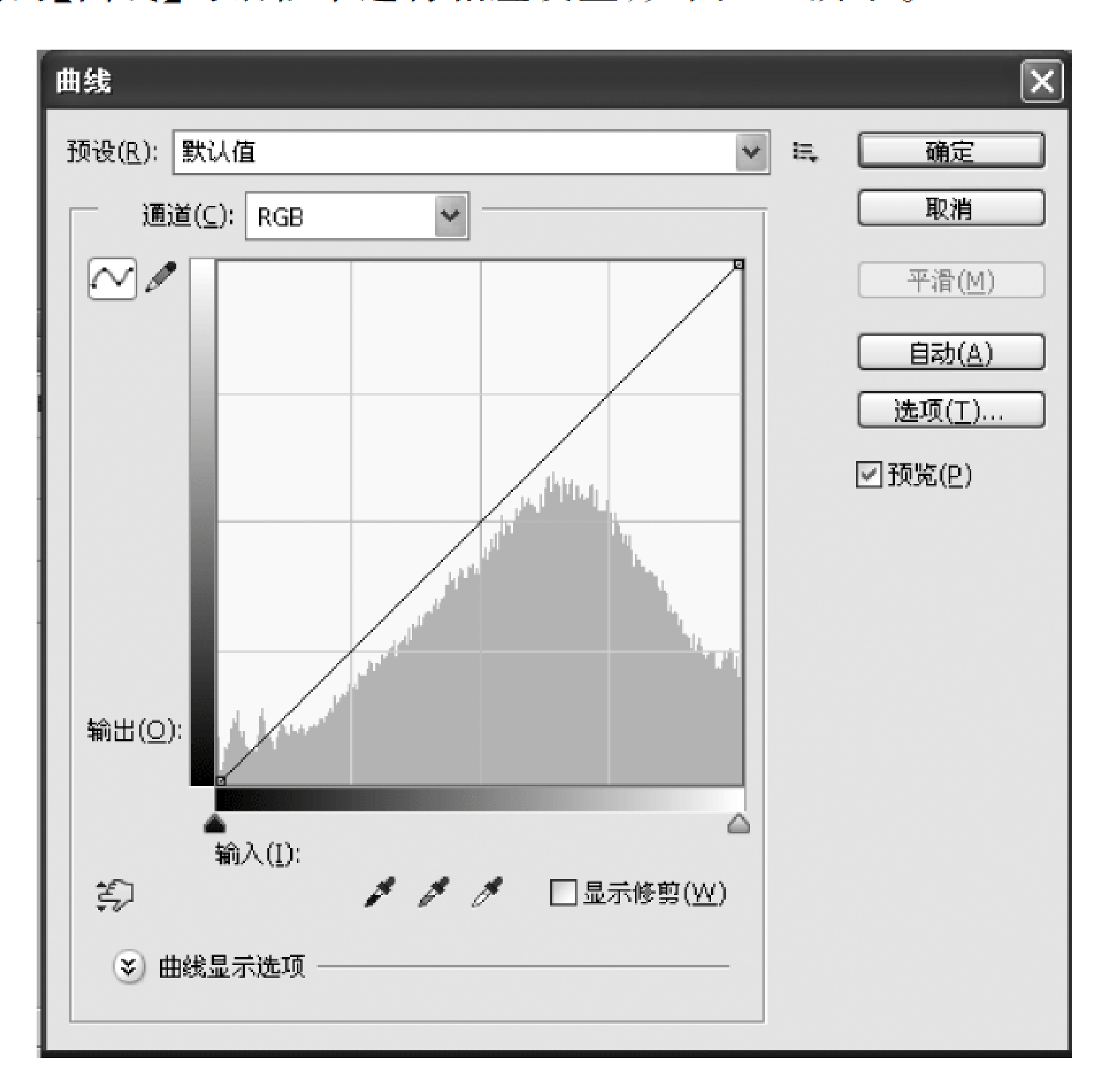


图 5-7 【曲线】对话框

- 水平轴表示像素原来的亮度值(输入值)，垂直轴表示新的亮度值(输出值)。
- 在曲线上单击可创建调节点，拖曳调节点即可调整图像中明暗的数量，由左下至右上为由暗至亮。
- 如果要删除调节点，只要选择该调节点，再按 Delete 键，或直接拖曳该调节点离开曲线。
- 单击【曲线】对话框中的铅笔按钮，可以通过直接在【曲线】对话框中绘制曲线来调整图像亮度。
- 在【显示数量】选项中选择光线总量或墨水总量。
- 在【显示】选项中选择是否显示通道、直方图、基线及交叉线等信息。

6.【色彩平衡】命令

【色彩平衡】命令主要用于更改图像的整体颜色，可以通过在彩色图像中改变颜色的混合比例来进行整图的色彩校正。选择【图像】|【调整】|【色彩平衡】命令，在弹出的【色彩平

衡】对话框进行相应设置，如图 5-8 所示。

- 在【色彩平衡】类参数下拖曳三角形滑块可以调整要在图像中增加或减少颜色。
- 在【色调平衡】类下选择【阴影】、【中间调】或【高光】单选按钮，可以设置色彩校正对图像的哪一部分起作用。
- 【保持亮度】：在对 RGB 图像进行操作时应勾选该复选框，以防止在更改颜色时更改了图像中的亮度值，此复选框可保持图像中的色调平衡。

7.【亮度/对比度】命令

使用【亮度/对比度】命令可以调整图像的亮度和对比度，其对话框如图 5-9 所示。



图 5-8 【色彩平衡】对话框



图 5-9 【亮度/对比度】对话框

8.【黑白】命令

使用【黑白】命令可以调整 6 种不同颜色(红、黄、绿、青、蓝、洋红)的亮度值，从而制作出高质量的黑白照片，还可以制作出各种不同颜色的单色照片，其对话框如图 5-10 所示。

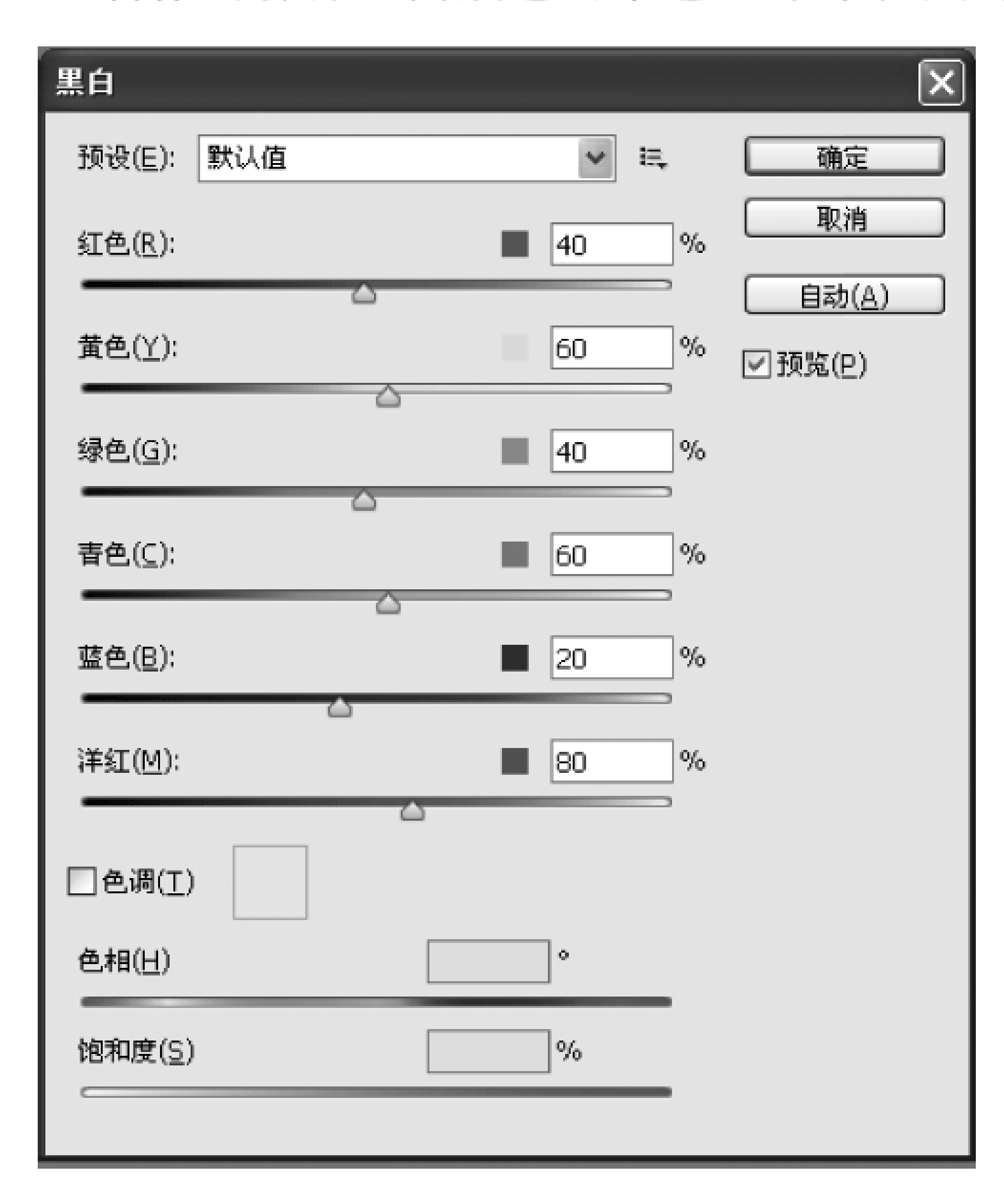


图 5-10 【黑白】对话框

- 【预设】：可以在此下拉列表中选择不同的预置方案，还可以自定义不同颜色的色值。
- 【色调】：勾选该复选框，可以分别调整色相和饱和度。

9.【色相/饱和度】命令

【色相/饱和度】命令主要用于调整图像中单个颜色成分的色相、饱和度和明度。选择菜单栏中的【图像】|【调整】|【色相/饱和度】命令，在弹出的对话框中进行相应设置，如图 5-11 所示。



图 5-11 【色相/饱和度】对话框

- 【编辑】：在此下拉列表中选择【全图】选项，可以调整图像的色调、饱和度和亮度，另外还可以在此下拉列表中选择其他特定颜色。
- 【色相】值决定调整的颜色。
- 调整【饱和度】和【明度】值或移动相应的滑块可以调整图像或选定颜色的饱和度和亮度。
- 【色相/饱和度】对话框的最下方有两条色带，在调整图像色调时，上面的色带颜色显示调整前的像素颜色，下面的色带颜色将置换与上面色带对应的颜色。
- 【着色】：勾选该复选框，可以使用同一种颜色置换原图中的颜色。

10.【去色】命令

使用【去色】命令可以丢弃图像色彩，使图像以灰色显示。但使用【去色】命令只是将图像中原有的色彩丢弃，并不是将图像的颜色模式修改为灰度模式。

11.【匹配颜色】命令

使用【匹配颜色】命令可以将两个图像或同一图像中两个图层的颜色和亮度相匹配，使其颜色和亮度协调一致。其中需要改变颜色和亮度的图像称为“目标图像”，需要采样的图像称为“源图像”，其对话框如图 5-12 所示。



图 5-12 【匹配颜色】对话框

12.【替换颜色】命令

使用【替换颜色】命令可以修改图像中特定的颜色。当图像窗口中存在图像文件时，选择【图像】|【调整】|【替换颜色】命令，在弹出的对话框中对颜色进行吸取（可进行选区的加减），设置容差，调整色相、饱和度、明度等值，将吸取的颜色用新颜色替换。【替换颜色】对话框如图 5-13 所示。



图 5-13 【替换颜色】对话框

13.【可选颜色】命令

使用【可选颜色】命令可以对指定的颜色进行精细调整，以校正不平衡问题。使用【可选颜色】命令校正颜色是使用高档扫描仪和分色程序时的一个技巧，可在图像中的每个加色（或减色）的原色成分中增加（或减少）印刷颜色的量。

选择【图像】|【调整】|【可选颜色】命令，在弹出的对话框中设置参数，如图 5-14 所示。

- 【颜色】：在此下拉列表中选择要调整的颜色，通过调整【青色】、【洋红】、【黄色】和【黑色】值可以修改指定的颜色。
- 选择【相对】单选按钮，可以按照总量的百分比更改现有青色、洋红、黄色或黑色的量。选择该单选按钮不能调整纯白色或纯黑色。

- 选择【绝对】单选按钮，则按绝对值调整颜色。例如，如果从 50%的洋红的像素开始，然后添加 10%，则洋红会设置为 60%。



图 5-14 【可选颜色】对话框



图 5-15 【通道混和器】对话框

14.【通道混和器】命令

【通道混和器】命令是使用当前颜色通道的混合来修改颜色通道，从而达到改变图像颜色的目的。

选择【图像】|【调整】|【通道混和器】命令，将弹出【通道混和器】对话框，其中的内容会根据图像模式的不同而产生相应的变化，这里以 RGB 图像为例，【通道混和器】对话框如图 5-15 所示。

- 【输出通道】：在此下拉列表中选择要进行调整的通道。
- 在【源通道】下移动滑块，可以调整源通道在输出通道中所占的百分比。
- 修改【常数】值可以将一个不同透明度的通道添加到输出通道中，负值作为黑色通道，正值作为白色通道。
- 【单色】复选框：勾选该复选框，对所有输出通道应用相同的设置，此时创建仅包含灰色值的彩色图像。

15.【渐变映射】命令

利用【渐变映射】命令可以使用指定的渐变填充颜色在图像中按图像灰度级由暗至亮取代原图的颜色。选择【图像】|【调整】|【渐变映射】命令，将弹出图 5-16 所示的对话框。

- 单击【灰度映射所用的渐变】颜色条，可以在弹出的【渐变编辑器】对话框中选择并编辑要使用的渐变项，如果所选择的渐变中有透明效果，则透明不起作用。
- 【仿色】：勾选该复选框，可以使渐变过渡更加均匀。
- 【反向】：勾选该复选框，将反转过渡项中的颜色顺序。



图 5-16 【渐变映射】对话框

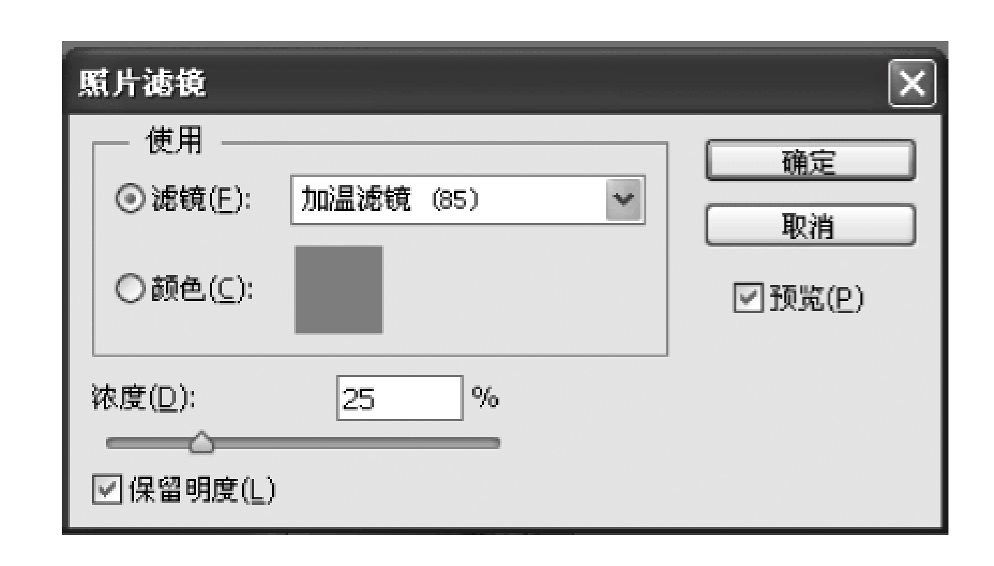


图 5-17 【照片滤镜】对话框

16.【照片滤镜】命令

【照片滤镜】命令模拟在相机镜头前面加彩色滤镜，以便调整通过镜头传输的光的色彩平衡和色温，使胶片曝光。选择【图像】|【调整】|【照片滤镜】命令，弹出【照片滤镜】对话框，如图 5-17 所示。

- 在【滤镜】下拉列表中可以选择 Photoshop 提供的预设滤镜效果。
- 选择【颜色】单选按钮，单击其右侧的色块，可以在弹出的【拾色器】对话框中设置滤镜的颜色。
- 【浓度】值决定应用的颜色对图像的影响程度。此值越大，滤镜颜色对图像的影响就越大。
- 【保留明度】：勾选该复选框，则滤镜颜色不影响图像的亮度。

17.【阴影/高光】命令

【阴影/高光】命令不是简单地使图像变暗或变亮，而是基于暗调或高光中的周围像素(局部相邻像素)增亮或变暗。该命令允许分别控制暗调和高光，适合用于校正由强逆光而形成阴影的照片，或者校正由于太接近相机闪光灯而有些发白的焦点。【阴影/高光】命令的默认值被设置为修复具有逆光问题的图像。

选择【图像】|【调整】|【阴影/高光】命令，弹出【阴影/高光】对话框，如图 5-18 所示。

- 【数量】值决定调整光照的校正量。此值越大，为图像中的暗调提供的增亮程度或者为高光提供的变暗程度越大。
- 【色调宽度】值控制暗调或高光中色调的修改范围。此值较小时，只对图像中的较暗或较亮区域进行校正调整；此值越大，包括的色调调整区域越多。
- 【半径】值决定每个像素周围多大的范围内的像素作为“周围像素”，再根据“周围像素”决定当前像素属于暗调还是高光。

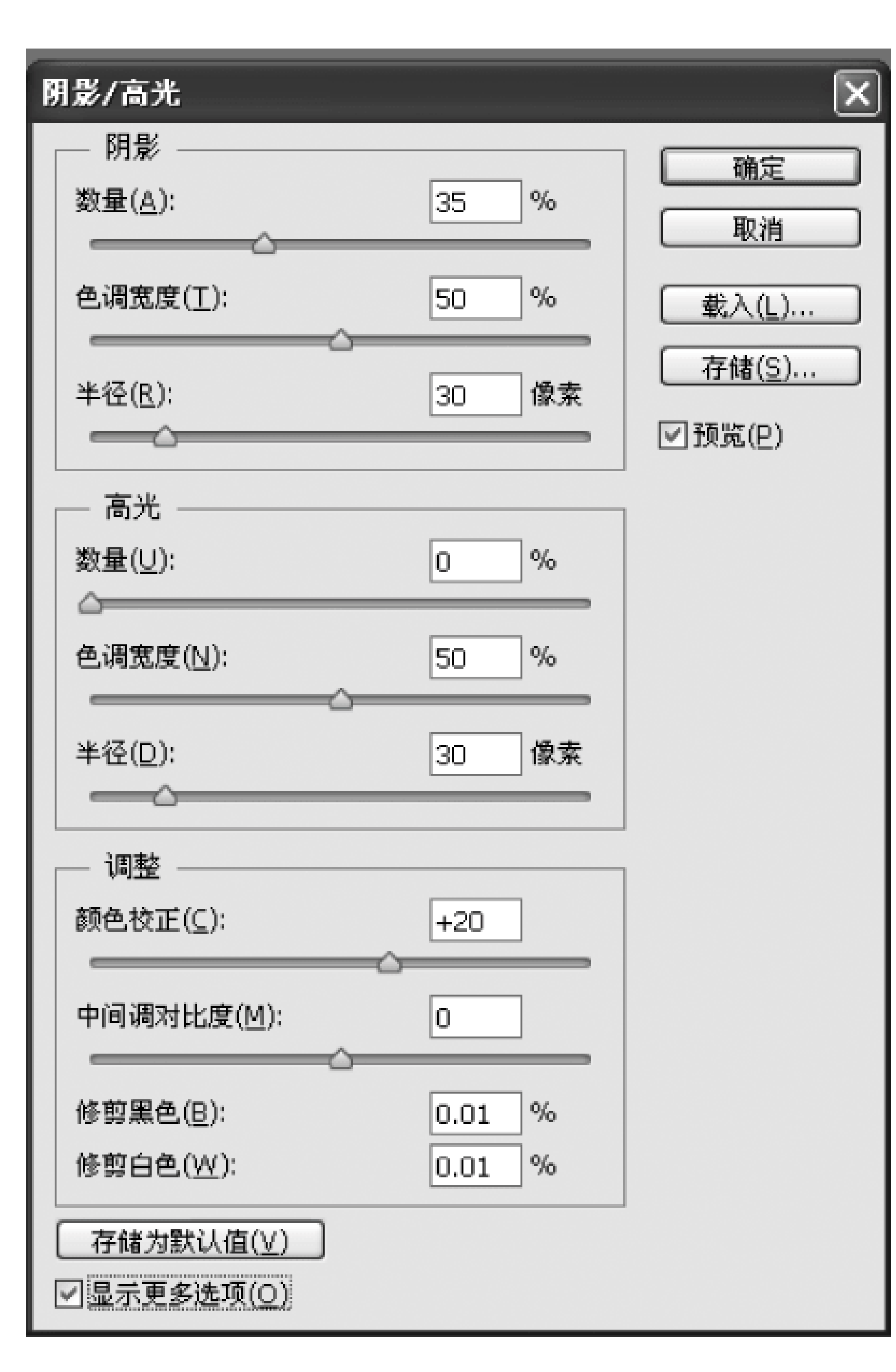


图 5-18 【阴影/高光】对话框

- 【颜色校正】值主要用于在图像已更改明暗的区域中微调颜色，此调整仅适用于彩色图像。通常此值越大，生成的颜色越饱和；此值越小，生成的颜色越不饱和。
- 【中间调对比度】值主要用于调整图像中间调的对比度。“中间调”是指图像中除明暗色调以外的中间色调。
- 【修剪黑色】和【修剪白色】值决定会将图像中多少比例的两端阶调(即极端阴影(色阶为“0”)和极端高光(色阶为“255”))舍弃掉，并将剩余部分的阶调拉开至“0～255”，从而加大图像的对比度。此值越大，生成的图像的对比度越大。注意不要将此值设置得太大，否则会减小阴影或高光的细节。

18.【曝光度】命令

使用【曝光度】命令可以调整 HDR(32 位)图像的色调，也可以调整 8 位和 16 位图像的色调。曝光度是通过在线性颜色空间(灰度系数为 1.0)而不是在图像的当前颜色空间执行计算得到的。选择【图像】|【调整】|【曝光度】命令，将弹出【曝光度】对话框。

19.【反相】命令

使用【反相】命令可以得到图像的反相效果，即图像中的颜色和亮度全部反转，转换为 256 级中相反的值。例如，如果原像素颜色的 R、G、B 值分别为“200”、“50”、“30”，那么反转后它的 R、G、B 值分别为“55”、“205”、“225”；如果原图像像素的亮度级为“100”，那么反转后该像素的亮度级为“155”。

20.【色调均化】命令

Photoshop 软件将每个通道中最亮和最暗的像素定义为白色和黑色，然后按比例重新分配中间像素值，使图像中的明暗分布均匀。

21.【阈值】命令

使用【阈值】命令可以将一个灰度或彩色图像转换为高对比度的黑白图像。此命令是将一定的色阶指定为阈值，所有比该阈值亮的像素会被转换为白色，所有比该阈值暗的像素会被转换为黑色。【阈值】对话框如图 5-19 所示。

22.【色调分离】命令

使用【色调分离】命令可以由用户指定图像中每个通道的色调级(或亮度值)的数目，并将图像像素映射到最接近的匹配色调上。例如，将 RGB 图像中的通道设置为只有两个色调，那么图像只能产生 6 种颜色，即两种红色、两种绿色和两种蓝色，其对话框如图 5-20 所示。

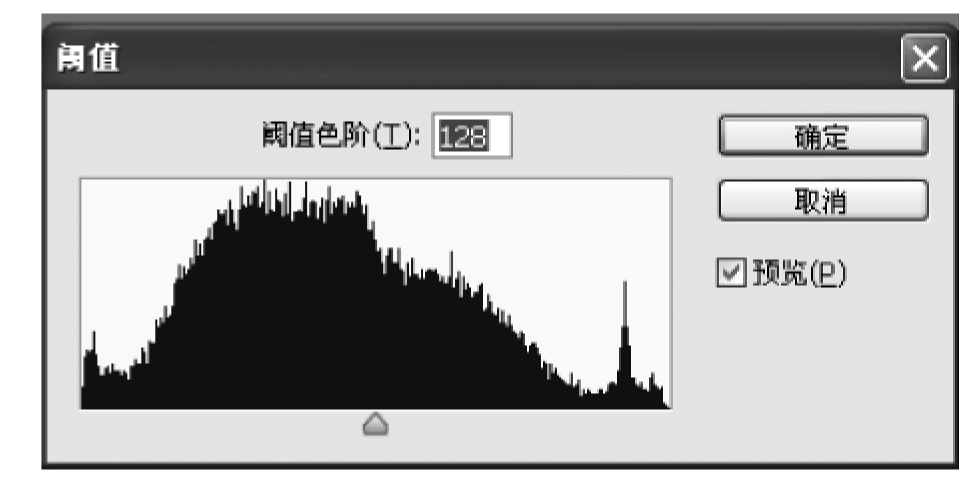


图 5-19　【阈值】对话框



图 5-20　【色调分离】对话框

23.【变化】命令

使用【变化】命令能可视地调整图像或选区的色彩平衡、对比度、亮度和饱和度，此命令对于不需要精确调整色彩的平均色调图像最有用，但该命令不能用在索引颜色图像上。选择【图像】|【调整】|【变化】命令，弹出【变化】对话框，如图 5-21 所示。

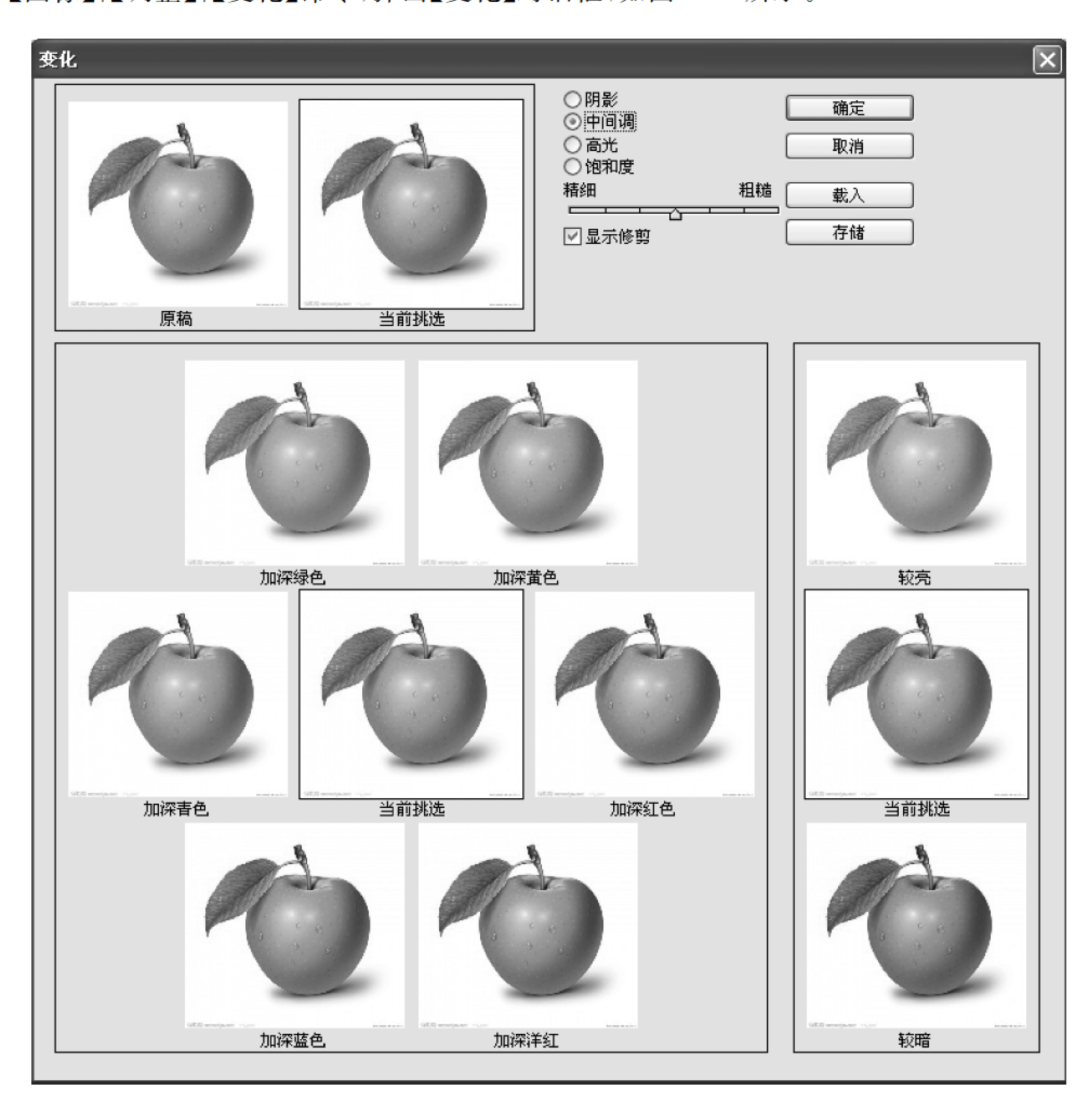


图 5-21 【变化】对话框

- 【原稿】缩览图显示原始图像的效果，【当前挑选】缩览图显示调整后的图像效果。
- 选择【阴影】、【中间调】或【高光】单选按钮可以调整图像的暗区、中间区域或亮区，选择【饱和度】单选按钮可以更改图像中的色相数。
- 【精细/粗糙】滑块可以设定每次调整的程度。
- 【变化】对话框左下方的 7 个加色缩览图显示当前图像增加某色后的效果，单击【变化】对话框右下方的 3 个缩览图可以调整图像的亮度。

5.3 滤镜

Photoshop 中的滤镜大体可以分为两组，第一组滤镜较复杂，在对话框中可以通过综合使用设置参数和选项、使用命令按钮及鼠标等产生特殊效果的图像，被称为高级滤镜，这组滤镜包括【转换为智能滤镜】命令、【滤镜库】命令、【液化】命令、【自适应广角】命令、【镜头校正】命令、【油画】命令和【消失点】命令；第二组滤镜是 Photoshop 的传统滤镜，也被称为标准滤镜，又分 12 大类，包括【像素化】滤镜、【扭曲】滤镜、【杂色】滤镜、【模糊】滤镜、【画笔描边】滤镜、【素描】滤镜、【纹理】滤镜、【艺术效果】滤镜、【视频】滤镜、【锐化】滤镜、【风格化】滤镜和【其他】滤镜。

另外，【滤镜】菜单的最后一项是 Digimarc，该组命令主要用于在图像中添加和读取水印(即图像信息)，以保护图像版权。

1.【转换为智能滤镜】命令

智能滤镜是一种非破坏性的滤镜，可以像使用图层样式那样随时调整滤镜参数、隐藏或者删除，这些操作都不会对图像造成任何实质性的破坏。

选中需要应用滤镜的图层，选择【滤镜】|【转换为智能滤镜】命令，将所选图层转换为智能对象，然后再使用滤镜，即可创建智能滤镜，在 Photoshop 中除了【镜头校正】、【液化】和【消失点】滤镜外，其他滤镜都可以作为智能滤镜使用。

2.【滤镜库】命令

在处理图像时，可能需要单独使用某一滤镜，或者使用多个滤镜，或者将某滤镜在某图像中应用多次。使用【滤镜库】命令不仅能轻松地一次性完成这几种设置，而且可以预览图像应用多重滤镜后的效果。

3.【液化】命令

【液化】命令主要是使图像产生特殊的扭曲效果。在【液化】对话框中可以在左侧的工具列表中选择扭曲工具，在右侧扭曲选项下设置参数，在图像中拖曳鼠标指针或按住鼠标左键不放进行扭曲操作。

4.【油画】命令

【油画】命令专门用来制作油画效果。

5.【消失点】命令

使用【消失点】命令可以在有透视角度的图像中进行图像编辑与处理。通过使用【消失点】命令来修饰、添加或移除图像中包含透视效果的内容。

6.【像素化】滤镜组

选择【滤镜】|【像素化】命令会弹出相应的子菜单，此类滤镜主要用来将图像分块或将图

像平面化。它并不是真正地改变了图像像素点的形状，只是在图像中表现出某种基础形状的特征，以形成一些类似像素化的形状改变。其中各滤镜的作用如下：

- 【彩块化】滤镜是通过分组和改变示例像素为相似的有色像素块生成手绘效果，或使现实主义图像变为抽象派绘画。此滤镜通过单击直接完成，不能人工调整参数控制其效果。
- 【彩色半调】滤镜是在图像中添加带有彩色半色调的网点，将图像中的每个颜色通道都转变成着色网点，网点的大小受其亮度影响。
- 【点状化】滤镜将图像分为随机的彩色斑点，空白部分由背景色填充。【点状化】滤镜的效果与【彩色半调】滤镜的效果相似，但【点状化】滤镜最终生成的是与原图像颜色一致的斑点，而不是各个通道的原色斑点。
- 【晶格化】滤镜是将图像中的像素分块，每块都使用同一种颜色，从而将原图像修改为以多边形的纯色色块组成的图像。
- 【马赛克】滤镜是通过将一个单元内的所有像素统一颜色来产生马赛克的效果。通常将此滤镜与【编辑】|【渐隐】功能结合使用，以得到理想的效果。
- 【碎片】滤镜是将图像复制为4份，再将它们平均和移位，从而形成一种不聚焦的“四重视”效果。【碎片】滤镜也是通过单击直接完成的，不能控制效果。
- 【铜版雕刻】滤镜是用点、线条和笔划重新生成图像，产生镂刻的版画效果。它在【类型】框中总共提供了10个笔型选项。

7.【扭曲】滤镜组

选择【滤镜】|【扭曲】命令会弹出相应的子菜单，此组滤镜主要是将当前图层或选区内的图层进行各种各样的扭曲变形。其中各滤镜的作用如下：

- 【波浪】滤镜是一种复杂的【扭曲】滤镜，也是一种精确的【扭曲】滤镜，它可以由用户控制波动的效果，常用来制作不规则扭曲的效果，如闪电、飘动的旗子、卷曲的纸等。
- 【波纹】滤镜产生水纹的涟漪效果，常被用来制作水面倒影等效果。
- 【极坐标】滤镜将图像坐标从平面坐标转换为极坐标，或从极坐标转换为平面坐标，它使图像产生一种极度变形的效果，经常把它作为图案设计工具或用它创建一些特殊效果的图像。
- 【挤压】滤镜在图像或选区中间生成一个向内或向外的凸起，它的效果有点像【球面化】滤镜，但它的凸起形态变形较严重。
- 【切变】滤镜可以根据在对话框中建立的曲线使图像产生弯曲效果。
- 【球面化】滤镜通过将选区折成球形、扭曲图像以及伸展图像以适合选中的曲线，使对象具有3D效果。它可以将图像中所选定的球形区域或其他区域扭曲膨胀或变形缩小，也可以在水平方向或垂直方向上进行单向球化。经常使用【球面化】滤镜制作带有折射效果的球体效果，如玻璃球、金属球等，也可以利用该滤镜制作圆柱形或圆柱形表面的效果，如饮料罐、笔筒、酒瓶上的标签等。
- 【水波】滤镜模拟水面上产生起伏旋转的波纹效果。
- 【旋转扭曲】滤镜产生旋转的风轮效果，旋转的中心为图像或选区的中心。

- 【置换】滤镜是根据另一个“.psd”格式图像的明暗度将当前图像中的像素移动，从而产生变形的效果。

8.【杂色】滤镜组

选择【滤镜】|【杂色】命令，会在弹出的子菜单中列出各滤镜，此类滤镜主要用于在图像中按一定方式混入杂点，以制作出着色像素图案的纹理。其中各滤镜的作用如下：

- 【减少杂色】滤镜可以去除图像中的杂色以及消除 JPEG 存储低品质图像导致的斑驳效果。
- 【蒙尘与划痕】滤镜的作用是搜索图像中的缺陷并将其融入到周围像素中。它将根据亮度的过渡差值找出突出于其周围像素的像素，并用周围的颜色填充这些区域，这是消除图像划痕的有效方法，但它也有可能将图像中应有的亮点清除，所以要慎重使用。
- 【去斑】滤镜将寻找图像中色彩范围变化最大的区域，然后模糊除去、过滤边缘外的东西，使图像中的噪音减少，这个滤镜没有选项，单击即可直接执行，它在清除图像杂点的同时会使图像产生一定的模糊，所以大家在使用时要慎重。
- 【添加杂色】滤镜用来将一定数量的杂色点以随机的方式引入到图像中，它可以使混合时产生的色彩具有漫散的效果。
- 【中间值】滤镜通过混合选区中像素的亮度来减少图像的杂色，此滤镜搜索像素选区的半径范围以查找亮度相近的像素、扔掉与相邻像素差异太大的像素，并用搜索到的像素中间亮度值替换中心像素，此滤镜在消除或减少图像的动感效果时非常有用。

9.【模糊】滤镜组

【模糊】滤镜组中滤镜的作用是将图像边缘过于清晰或对比度过于强烈的区域进行模糊，以产生各种不同的模糊效果，使图像看起来更朦胧一些。其中各滤镜的作用如下：

- 【场景模糊】滤镜允许用户创建一个简单的全局模糊效果，或从 A 点到 B 点的局部模糊(以继续添加 C、D、E 等各点)。当用户应用这个滤镜后，会在图像中出现一个单独的大头钉标记，可以通过它对模糊进行控制。
- 【光圈模糊】滤镜用于将镜头的焦点对准图像中的某一特定对象，这个滤镜会在需要聚焦的对象周围创建一个变亮的椭圆区域，用户可以对这个椭圆的大小进行调节。
- 【倾斜偏移】滤镜可以让图片产生类似于微缩景观的效果。然而，并不是所有的图像都适合制作这种效果。通常情况下，选择主题离镜头较远的俯拍图。
- 【表面模糊】滤镜可以在保留边缘的同时模糊图像，此滤镜常用于创建特殊效果并消除杂色或粒度。
- 【动感模糊】滤镜只在单一方向上对图像像素进行模糊处理。它可以产生动感模糊的效果，模仿物体高速运动时曝光的摄影手法，一般较适合运动物体处于画面中心、周围背景变化较少的图像。
- 【方框模糊】滤镜只有【半径】一个选项，该值决定每个调整区域的大小，此值越大，图像越模糊。

- 【高斯模糊】滤镜是 Photoshop 中较常使用的滤镜之一，它依据高斯曲线来调节图像的像素色值。【高斯模糊】滤镜只有一个【半径】选项，调整【半径】值可以控制模糊程度，从较微弱的模糊直到造成难以辨认的浓厚的图像模糊。
- 【进一步模糊】滤镜产生一个固定的较弱的模糊效果，它与前面的【模糊】滤镜效果相似，但其模糊程度是【模糊】滤镜的 3～4 倍。
- 【径向模糊】滤镜可以创建一种旋转或放射模糊的效果。
- 【镜头模糊】滤镜向图像中添加模糊以产生更窄的景深效果，以便使图像中的一些对象在焦点内，而使另一些区域变模糊。例如可以将照片中的前景保持清晰，而使背景变得模糊。
- 【模糊】滤镜可以产生较轻微的模糊效果，它的模糊效果是固定的，常用它来模糊图像边缘。
- 【平均】滤镜就是找出图像或选区中的平均颜色，然后用该颜色填充图像或选区。
- 【特殊模糊】滤镜可以产生一种清晰边界的模糊效果，它只对有微弱颜色变化的区域进行模糊，不对边缘进行模糊。也就是说，该滤镜能使图像中原来较清晰的部分不变，而使原来较模糊的部分更加模糊。
- 【形状模糊】滤镜可以以一定形状为基础进行模糊处理。

10.【渲染】滤镜组

【渲染】滤镜组中的滤镜主要用于改变图像的光感效果，例如模拟在图像场景中放置不同的灯光，可以产生不同的光源放果，也可以与通道相配合产生一种特殊的三维浮雕效果。【渲染】滤镜组中各滤镜的作用如下：

- 【分层云彩】滤镜的效果与原图像的颜色有关，它并不完全覆盖图像，而是相当于在图像中添加了一个差异色云彩效果。
- 【镜头光晕】滤镜将产生摄像机镜头光晕效果，并可自动调节摄像机光晕的位置，可以创建星光效果、强烈的日光效果以及其他光芒效果等。
- 【纤维】滤镜使用前景色和背景色创建编织纤维的外观，通常用来制作纤维织品的效果。
- 【云彩】滤镜是利用前景色和背景色随机组合将图像转换为柔和的云彩效果，单击即可直接执行。使用【云彩】滤镜会将原图像全部覆盖。

11.【画笔描边】滤镜组

【画笔描边】滤镜组中的滤镜主要利用不同的画笔和油墨描边效果创造出艺术绘画效果。该滤镜组中的所有滤镜都在 Photoshop 的滤镜库中。【画笔描边】滤镜组中各滤镜的作用如下：

- 【成角的线条】滤镜产生一种不一致方向的倾斜笔触效果，笔触的方向在图像的不同颜色区域内发生变化，用一个方向的线条绘制图像的亮区，用相反方向的线条绘制暗区。
- 【墨水轮廓】滤镜是以钢笔画的风格用纤细的线条在原图细节上重绘图像。
- 【喷溅】滤镜是在图像中产生画面颗粒飞溅的效果，就好像用水喷在图像上，使图像中出现一些细微的颜料滴。

- 【喷色描边】滤镜的效果与【喷溅】滤镜的效果有些相似，但它产生的是倾斜飞溅效果，有时利用【喷色描边】滤镜制作下雨的效果。
- 【强化的边缘】滤镜主要用于强化图像中不同颜色之间的边界，使图像产生一种强调边缘的效果。
- 【深色线条】滤镜生成的也是交叉笔触，它用短的、绷紧的线条绘制图像接近黑色的暗区，用长的白色线条绘制图像中的亮区。该滤镜可以使图像产生一种很强烈的黑色阴影。
- 【烟灰墨】滤镜是以日本画的风格绘画，看起来像是用蘸满黑色油墨的湿画笔在宣纸上绘画，这种效果具有非常黑的柔化模糊边缘，该滤镜通过计算图像像素的色值分布来产生色值概括描绘效果。
- 【阴影线】滤镜保留原始图像的细节和特征，同时使用模拟的铅笔阴影线添加纹理，并使彩色区域的边缘变暗。

12. 【素描】滤镜组

【素描】滤镜组中的大部分滤镜将以前景色和背景色置换原图中的色彩，产生一种精确的图像效果，通常用于获得 3D 效果或创建精美的艺术品和手绘外观。【素描】滤镜组中共有 14 种滤镜，都被保存在 Photoshop 的滤镜库中，只要打开滤镜库，就可以方便地查看和设置每个【素描】滤镜。在介绍【素描】滤镜组时，如无特别说明，则将前景色设为黑色，将背景色设为白色。

【素描】滤镜组中各滤镜的作用如下：

- 【半调图案】滤镜是使用前景色和背景色的组合重新给图片上色，产生一种网板图案的效果。
- 【便条纸】滤镜是根据图像中像素的明暗用前景色和背景色替换原图中像素的颜色，使图像产生一种类似用厚纸制作的有凹陷效果的作品。
- 【粉笔和炭笔】滤镜合成背景颜色的粉笔笔触和前景颜色的炭笔笔触，产生一种用粉笔和炭笔涂抹的草图效果，与【编辑】|【渐隐粉笔和炭笔】命令结合使用会取得更好的效果。
- 【络黄】滤镜产生的效果有点像【塑料效果】滤镜，只是它生成的效果是灰度的，原图像的细节几乎全部丢失，产生一种液体金属的质感，有时会用这一滤镜配合【渐隐】命令表现冰块、玻璃、水面或绸缎等表面光滑的效果。
- 【绘图笔】滤镜是用前景色和背景色生成一种钢笔画素描的效果，图像中没有轮廓，只有一些具有细微变化的笔触。
- 【基底凸现】滤镜在图像中产生一种浮雕效果，图像用前景色和背景色填充。由于使用这一滤镜的图像细节丢失较多，所以常将其与菜单栏中的【编辑】|【渐隐基底凸现】命令配合使用。
- 【水彩画纸】滤镜模仿在潮湿的纸张上作画的效果，它模糊颜色并减少图像反差，产生画面浸湿、扩散的效果。
- 【撕边】滤镜是用前景色和背景色填充图像，并在前景色与背景色的交界处制作溅射的效果，这一滤镜有时会被用来制作毛边效果。

- 【塑料效果】滤镜利用前景色和背景色填充图像，产生一种光滑的浮雕效果，其中图像的高光部分使用背景色、较暗部分使用前景色。
- 【炭笔】滤镜模拟炭笔素描的效果，图像中较暗的区域用前景色的炭笔笔触着色，较亮的部分用背景色填充。
- 【炭精笔】滤镜生成一种用蜡笔笔触以前景色和背景色在花纹纸上描绘的效果，常用这种工具配合【编辑】|【渐隐炭精笔】命令制作磨损的地毯、麻布等效果。
- 【图章】滤镜用前景色和背景色填充图像，产生的效果类似于影印，但没有影印清晰，是一种类似于用图章盖印双色图像的效果。
- 【网状】滤镜也是用前景色和背景色填充图像并在图像上产生不规则的噪声，从而生成一种网状覆盖的效果。这一滤镜常被用来表示磁砖等建筑材料的效果。
- 【影印】滤镜产生的效果就像是在破旧的复制机上复制图像一样，其色彩由前景色和背景色填充，图像模糊、不均匀，并且有色调分离的效果，常用这一滤镜制作旧照片的效果。

13.【纹理】滤镜组

【纹理】滤镜组中滤镜的主要功能是使图像产生各种纹理过渡的变形效果，常用来创建图像的凹凸纹理和材质效果。其中各滤镜的作用如下：

- 【龟裂缝】滤镜是将浮雕效果与某种爆裂效果结合，产生凹凸不平的裂缝效果。
- 【颗粒】滤镜产生表面颗粒的效果，它相当于一个可控制的【添加杂色】滤镜，常用这一滤镜制作破损或脏旧的效果。
- 【马赛克拼贴】滤镜产生不规则的、近似方形马赛克瓷砖的效果。
- 【拼缀图】滤镜产生建筑拼贴瓷片的效果，其中原图中较暗的部分拼贴瓷片的高度较低，较亮的部分拼贴瓷片的高度较高。
- 【染色玻璃】滤镜产生从背后被照亮的不规则分离的“彩色玻璃格子”的效果，它们之间用前景色填充，用背景色填充间隔，“玻璃格子”的色彩分布与图片中的颜色分布有关。这一滤镜常被用来制作玻璃窗、昆虫翅膀及龟裂的土地等。
- 【纹理化】滤镜是在图像中添加软件给出的纹理效果，或根据另一个文件的亮度值向图像中添加纹理，常用该滤镜制作布的效果。

14.【艺术效果】滤镜组

【艺术效果】滤镜组用来对图像进行艺术效果处理，此组滤镜只应用于RGB模式和多通道模式图像，可以使图像产生精美艺术品般的效果。其中各滤镜的作用如下：

- 【壁画】滤镜可产生古壁画的斑点效果，它往往用于在图像边缘上添加黑色边缘，并增加反差和饱和度。
- 【彩色铅笔】滤镜是模拟彩色铅笔在纯色背景上绘制图像的效果，图像中较明显的边缘被保留并带有粗糙的阴影线外观。
- 使用【粗糙蜡笔】滤镜的图像看上去好像是用彩色粉笔在带纹理的背景上描过边，在亮色区域，粉笔看上去很厚，几乎看不见纹理，在深色区域，粉笔似乎被擦去了，使纹理显露出来。
- 【底纹效果】滤镜是根据纹理的类型和色值产生一种纹理喷绘的效果，经常将它与菜

单栏中的【编辑】|【渐隐】命令结合使用来创建一种布料的效果或油画的效果。

- 【调色刀】滤镜使相近的颜色融合产生大写意的笔法效果。
- 【干画笔】滤镜模拟使用干画笔技术(介于油画和水彩画之间的绘画技术)绘制图像的边缘,使画面产生一种不饱和、不湿润、干枯的油画效果,相当于在一幅油画颜料未干时用一把干刷子在图画上涂抹的效果。
- 【海报边缘】滤镜主要是减少图像中的颜色数量并用黑色勾画轮廓,从而将图像转换成一种美观的招贴画效果。
- 【海绵】滤镜产生画面浸湿的效果,它模拟用海绵涂抹的效果。有时用此滤镜表现水渍的效果。
- 【绘画涂抹】滤镜就像是一系列滤镜效果的共同作用,先将图像柔化,再描边,然后进行色调分离处理,最后锐化得到。
- 【胶片颗粒】滤镜产生一种软片颗粒纹理效果,在增加图像噪声的同时增亮图像并加大其反差。
- 【木刻】滤镜是对图像中的颜色进行色调分离处理,得到几乎不带渐变的简化图像,处理结果类似于木刻画。
- 【霓虹灯光】滤镜通过结合前景色和背景色给图像重新上色,并产生各种彩色霓虹灯光的效果。利用【霓虹灯光】滤镜可以创建出许多神奇而美丽的效果。
- 【水彩】滤镜产生的是一种水彩画的效果,但它生成的色彩通常比一般常见的水彩画要深。
- 【塑料包装】滤镜增加图像中的高光并强调图像中的线条,使图像产生一种表面质感很强的塑料压膜效果。
- 【涂抹棒】滤镜看起来就像是模糊笔触产生的一种条状涂抹效果。

15.【视频】滤镜组

【视频】滤镜组主要包括【NTSC 颜色】和【逐行】两个滤镜。此组滤镜属于 Photoshop 的外部接口程序,用来从摄像机输入图像或将图像输入到录像带上。这两个滤镜只有当图像要在电视或其他视频设备上播放时才会用到。

- 【NTSC 颜色】滤镜转换图像中的色域,使之适合 NTSC(National Television StandardsCommittee,国家电视系统委员会制式)视频标准色域,以便图像可被电视机接收。
- 【运行】滤镜是通过消除图像中的异常交错线来平滑影视图像,它删除从视屏捕捉的图像上的横向扫描线,利用复制或内插法置换失去的像素。

16.【锐化】滤镜组

【锐化】滤镜组中的滤镜主要通过增加相邻像素点之间的对比度使图像清晰化。其中各滤镜的作用如下:

- 【USM 锐化】滤镜是【锐化】类滤镜中应用最多的一种滤镜,也是【锐化】类滤镜中唯一可控制效果的滤镜。该滤镜在处理过程中使用模糊的遮罩,以产生边缘轮廓锐化的效果,它可以在尽可能少地增加噪声的情况下提高图像的清晰度。

- 【进一步锐化】滤镜相当于连续多次使用下面的【锐化】滤镜，从而得到一种强化锐化的效果，提高图像的对比度和清晰度。
- 【锐化】滤镜作用于图像的全部像素，增加图像像素间的反差，对调节图像的清晰度起到一定的作用，但重复过多的锐化会使图像粗糙。
- 【锐化边缘】滤镜的作用与【锐化】滤镜的作用相似，但它仅仅锐化图像的轮廓部分，以增加不同颜色之间的分界。
- 【智能锐化】滤镜具有【USM 锐化】滤镜所没有的锐化控制功能，具有可设置的锐化计算方法，并可单独控制图像阴影和高光的锐化程度。

17.【风格化】滤镜组

【风格化】滤镜组中的滤镜主要通过置换像素和通过查找并增加图像的对比度产生印象派或其他风格化画派作品的效果。其作用如下：

- 【查找边缘】滤镜搜索主要色彩的变化区域，强化其过渡像素，产生用彩色铅笔勾描轮廓的效果。此滤镜直接执行，没有可调参数，常将此与【编辑】|【渐隐】命令共同使用，生成铅笔素描的效果。
- 【等高线】滤镜是在图像中围绕每个通道的亮区和暗区边缘勾画轮廓线，从而产生三原色的细窄线条，经常在这种效果的基础上修改或制作卡通、漫画等。
- 【风】滤镜是按图像边缘中的像素颜色增加一些小的水平线产生起风的效果，此滤镜不具有模糊图像的效果，它只影响图像的边缘。常用此滤镜来创建火焰、冰雪和高速运动的效果。
- 【浮雕效果】滤镜通过勾画图像或选区的轮廓和降低周围色值来生成浮雕的效果。
- 【扩散】滤镜创建一种分离模糊效果，看起来有点像透过磨砂玻璃看图像的效果。
- 【拼贴】滤镜是将图像分裂成指定数目的方块，并将这些方块从原位置移动一定的距离。此滤镜没有预览功能，所以可能需要多次调试。
- 【曝光过度】滤镜产生图像正片和负片混合的效果，类似于摄影中增加光线强度产生的过度曝光效果，此滤镜直接执行，没有可调参数。
- 【凸出】滤镜就是将图像附着在一系列的三维立方体或方锥体上。

18.【其他】滤镜组

【其他】滤镜组中各滤镜的作用如下：

- 【高反差保留】滤镜可以删除图像中亮度逐渐变化的部分，并保留图像中色彩变化最大的部分，它起到一定的压平图像颜色的作用。
- 【位移】滤镜根据对话框中的值进行图像偏移，常用它来编辑无缝图案中的单元图像。
- 用户可以通过【自定】滤镜创建过滤器，如清晰化、模糊化和浮雕等效果。
- 【最大值】滤镜通过提亮暗区边缘的像素将图像中的亮区放大，消减暗区。
- 【最小值】滤镜通过加深亮区的边缘像素将暗区放大，消减亮区。

19. Digimarc（作品保护）滤镜组

Digimarc（作品保护）滤镜组主要包括【嵌入水印】和【读取水印】两个滤镜，此组滤镜用

于添加作品标记到图像中。

- 【读取水印】滤镜将检查图像，看它是否有水印。如果没有水印，就会弹出一个【未发现水印】提示框，提示没有水印；如果有水印，就会显示出创建者的信息。
- 【嵌入水印】滤镜是在图像中加入识别图像创建者的水印，用这一功能要先获得一个ID，这个 ID 号是付钱给 Digimarc Corporation 后收到的号码。一旦有了这个号码，就可以单击【嵌入水印】对话框中的【确认】按钮，并根据对话框的提示，一步一步将个人的信息加入到图像中。

20. 自适应广角

对于摄影师以及喜欢拍照的摄影爱好者来说，拍摄风光或者建筑必然要使用 EF 16-35mm F2.8L II 或类似焦距的广角镜头进行拍摄。在用广角镜头拍摄照片时都会有镜头畸变的情况，让照片边角位置出现弯曲变形，即使再昂贵的镜头也是如此。

在最新的 Photoshop CS6 的滤镜菜单中添加了一个全新的【自适应广角】命令，该命令可以在使用 Photoshop CS6 处理广角镜头拍摄的照片时对镜头产生的变形进行处理，得到一张完全没有变形的照片。

21. 镜头校正

【镜头校正】滤镜根据各种相机与镜头的测量自动校正，可以轻易地消除桶状和枕状变形、照片周围的暗角，以及造成边缘出现彩色光晕的色相差。

5.4 综合应用

5.4.1 鲜花字

制作鲜花字的操作步骤如下：

(1) 选择菜单栏中的【文件】|【打开】命令，打开图 5-22 所示的背景图片。

(2) 按 Ctrl+A 键选中图片，按 Ctrl+C 键复制图片，然后选择菜单栏中的【图像】|【画布大小】命令，在打开的对话框中做相应设置，如图 5-23 所示。



图 5-22　打开背景图片



图 5-23　画布大小设置

（3）按 Ctrl＋V 键粘贴图片，并对新复制的图片按 Ctrl＋T 键执行自由变换，将水平设为“－100”，即做成镜面效果，移动好位置，效果如图 5-24 所示。

（4）单击【图层】面板下方的【创建新的填充或调整层】按钮，选择【色彩平衡】命令，将作为背景的图片的颜色调整为“偏红色”，效果如图 5-25 所示。



图 5-24　图片镜面效果



图 5-25　色彩平衡设置

（5）选择工具箱中的文字工具，录入文字“鲜花字”，并将“素材 2”、“素材 3”、“素材 4”打开，然后将里面的鲜花部分选中复制到现有文件中，并调整大小、位置，再选中这 3 个花的图层，按 Ctrl＋E 键合并图层，效果如图 5-26 所示。

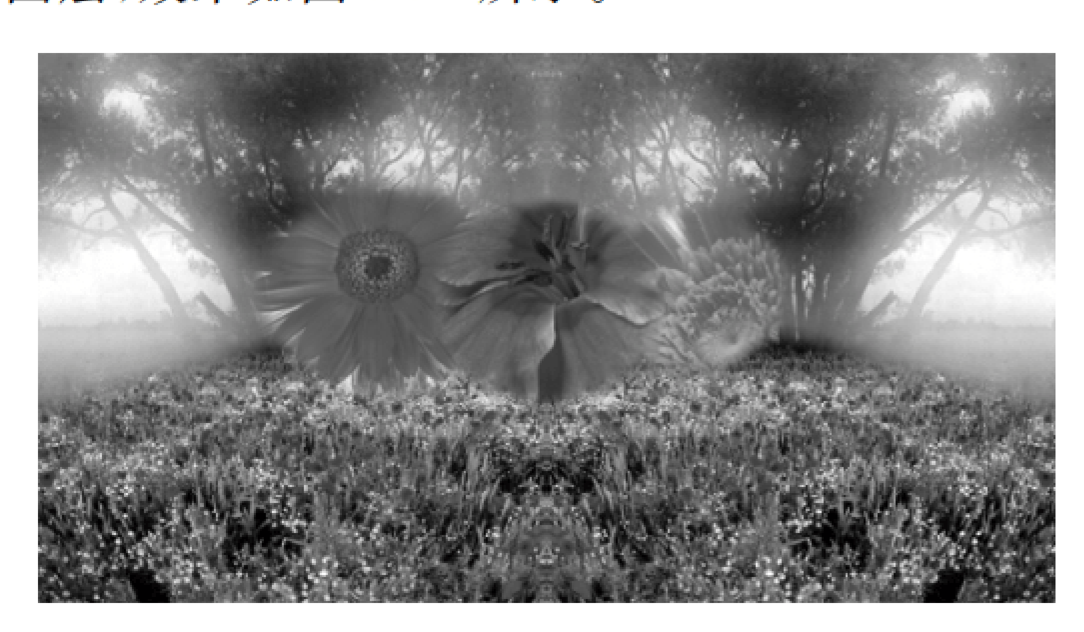


图 5-26　复制鲜花并合并图层

（6）按住 Ctrl 键，单击文字层的图层缩略图，将文字调入选区，然后选中花的图层，单击【图层】面板下面的【添加蒙版】按钮，如图 5-27 所示。



图 5-27　给文字添加蒙版

（7）单击【图层】面板下方的【新建图层】按钮，然后选中新图层，按 Ctrl 键单击文字缩略图，将文字调入选区，并选择【编辑】|【描边】命令，在对话框中设置参数，如图 5-28 所示。

（8）最终效果如图 5-29 所示，保存文件。





图 5-28　描边参数设置



图 5-29　最终效果

5.4.2　调色

给图片调色的操作步骤如下：

（1）选择菜单栏中的【文件】|【打开】命令，打开“素材 1”图形文件，如图 5-30 所示。

（2）按 Ctrl+J 键，复制原始素材，然后选择菜单栏中的【图像】|【调整】|【去色】命令，得到一张完全无色的黑白照片，如图 5-31 所示，接着将黑白图层复制。

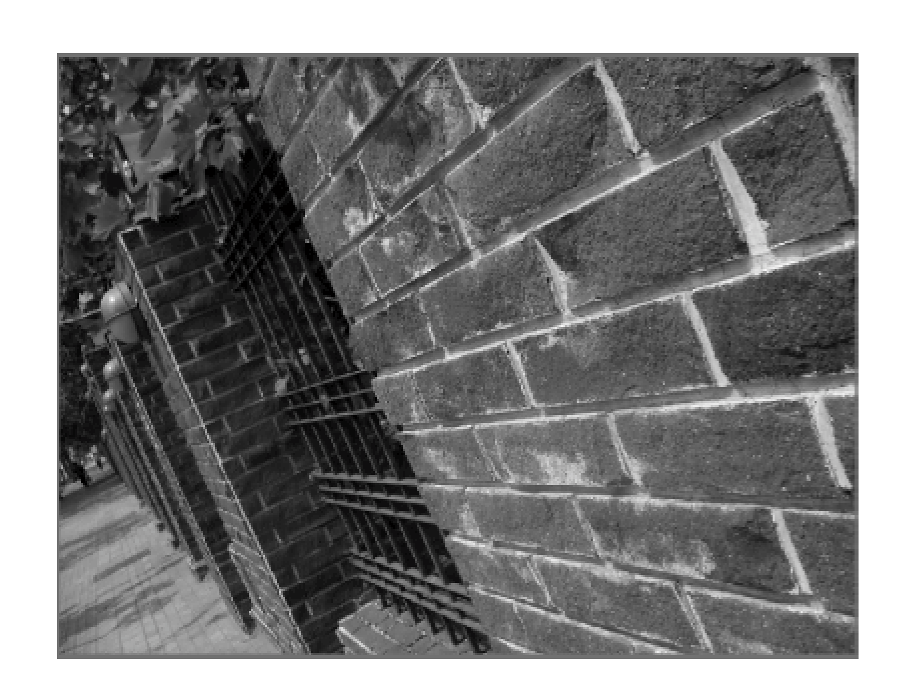


图 5-30　打开文件

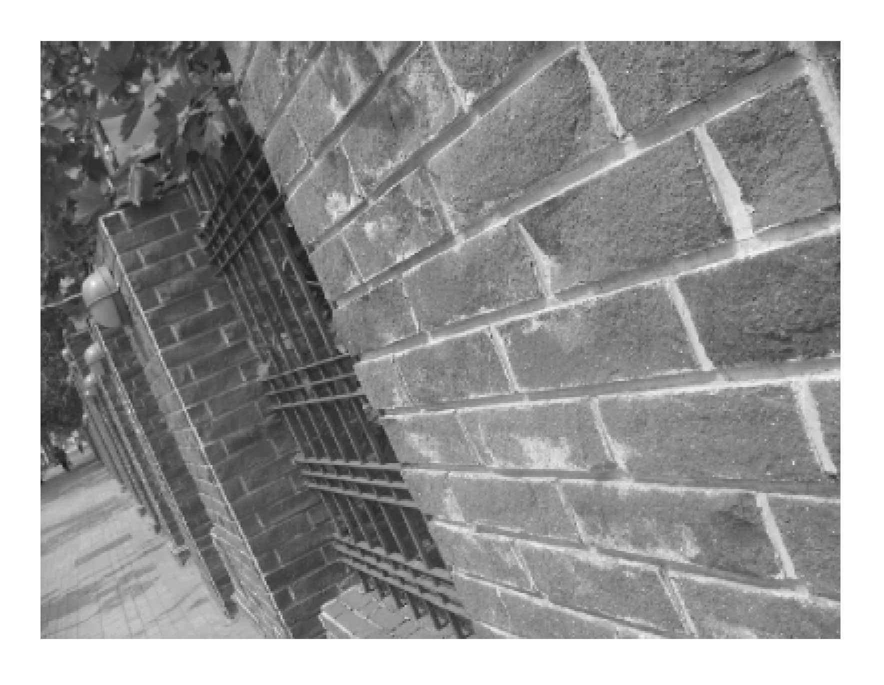


图 5-31　为图片去色

（3）选择位于背景层上方的黑白照片图层，选择【图像】|【调整】|【阈值】命令，在【阈值】对话框中设置参数如图 5-32 所示，单击【确定】按钮退出。

（4）选取最上方的黑白照片，选择【图像】|【调整】|【色调分离】命令，在【色调分离】对话框中设置参数如图 5-33 所示，单击【确定】按钮退出。

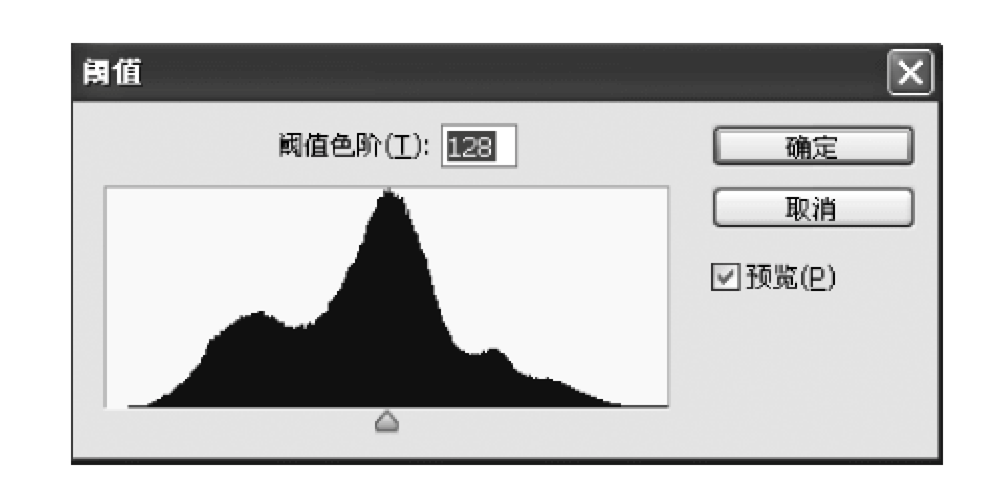


图 5-32 阈值参数设置



图 5-33 色调分离参数设置

(5) 将两个应用不同色调效果的黑白照片图层混合，然后选中最上面的图层，选择【图层混合】模式中的【正片叠底】，效果如图 5-34 所示。

(6) 在所有图层上建立一个新的图层，并填充为黄色(R：193，G：191，B：2)，然后将该图层的混合模式设为【正片叠底】，得到最终效果，如图 5-35 所示。



图 5-34 正片叠底后的效果



图 5-35 最终效果

5.4.3 时钟

制作时钟变换效果的操作步骤如下：

(1) 选择菜单栏中的【文件】|【打开】命令，打开“素材 1”图片，如图 5-36 所示。



图 5-36 打开文件

(2) 选择菜单栏中的【图像】|【调整】|【色相/饱和度】命令，打开【色相/饱和度】对话框，设置参数如图 5-37 所示，单击【确定】按钮，图像将变成具有陈旧感的黄色。

(3) 选择菜单栏中的【文件】|【打开】命令，打开时钟图片。然后使用魔棒工具，将容差设为“50”，按 Shift 键多次选择背景色，按 Ctrl+Shift+I 键反选出时钟，如图 5-38 所示。

(4) 使用移动工具将时钟复制到背景图片中，使用 Ctrl+T 键进行自由变换，将时钟调整为合适的大小，效果如图 5-39 所示。

(5) 选择菜单栏中的【滤镜】|【液化】命令，打开【液化】对话框，使用变形工具在时钟的下部进行拖动，使时钟变形，如图 5-40 所示，然后单击【确定】按钮退出。



图 5-37　【色相/饱和度】对话框参数设置



图 5-38　选择时钟



图 5-39　调整时钟

(6) 按 Ctrl+T 键打开自由变换调节框,按 Ctrl+Shift+Alt 键调整左侧的控点,从而调整时钟的透视效果,如图 5-41 所示。

(7) 使用多边形套索工具选取时钟上部,按 Ctrl+Shift+J 键将图像剪切到新的图层,如图 5-42 所示。

(8) 按 Ctrl+T 键打开自由变换调节框,按住 Ctrl 键调整控制点,使图像与柜子的面板在一个平面上,此时的效果如图 5-43 所示。



图 5-40 液化时钟



图 5-41 时钟透视效果



图 5-42 剪切时钟



图 5-43 自由变换时钟

（9）选择菜单栏中的【滤镜】|【液化】命令，打开【液化】对话框，调整时钟上面部分的变形效果，最终效果如图 5-44 所示。



图 5-44 最终效果

第6章 Flash CS6 概述

本章学习目标：

✍了解 Flash 窗口的组成。

✍掌握 Flash 中文件及窗口的操作。

✍掌握 Flash 动画的制作流程。

6.1 Flash CS6 简介

Flash CS6 以流控制技术和矢量技术为代表，能够将矢量图、位图、音频、动画和深一层交互动作有机地、灵活地结合在一起，从而制作出美观、新奇、交互性更强的动画效果。利用它制作出来的动画具有短小、精悍的特点。

Flash 的前身叫 FutureSplash，当时 FutureSplash 最大的两个用户是 Microsoft 和 Disney。

FutureSplash 最初是为 Netscape 开发的全新网页浏览插件。FutureWave 公司本来是打算把这项技术卖给 Adobe 公司，但是在那个时候 Adobe 公司根本不感兴趣，而 Macromedia 公司对此非常感兴趣，就这样在 1996 年的 12 月 FutureSplash 被正式卖给 Macromedia 公司，改名为 Flash1.0。后来，Macromedia 公司把 FutureSplash 重新命名为 Flash Player1.00。2005 年 4 月，Adobe 公司以 34 亿美元收购了 Macromedia。从此 Flash 开始了新一轮的改革，Adobe 发布了新的 SWF 开发平台 Flex。Flex 是面向传统程序员的 Flash 开发工具，Flex 基于 IBM Eclips，包含有比 Flash 更强劲的矢量用户界面组建系统和全新的 XML 扩展标记语言，开发者使用 Flex 可以快速部署被称为下一代互联网核心应用的 RIA(Rich Internet Application)系统，设计有良好用户体验和丰富交互特性的网站。与 Flash 相比，Flex 更适用于开发大型项目，并且更加注重代码编写。至 2010 年 Flash CS 5.0 发布，它可以和 Flash Builder(即最新版本的 Flex Builder)协作来完成项目。

6.2 Flash CS6 的工作环境

(1) 在启动 Flash CS6 之后会出现 Flash CS6 的开始页，如图 6-1 所示，这是许多 Macromedia 应用程序的标准页面，要想显示/隐藏此页面，可以选择【编辑】|【首选参数】命令，然后在打开的对话框中选择【常规】选项卡，在【启动时】下拉菜单中选择相应选项，如

图 6-2 所示。

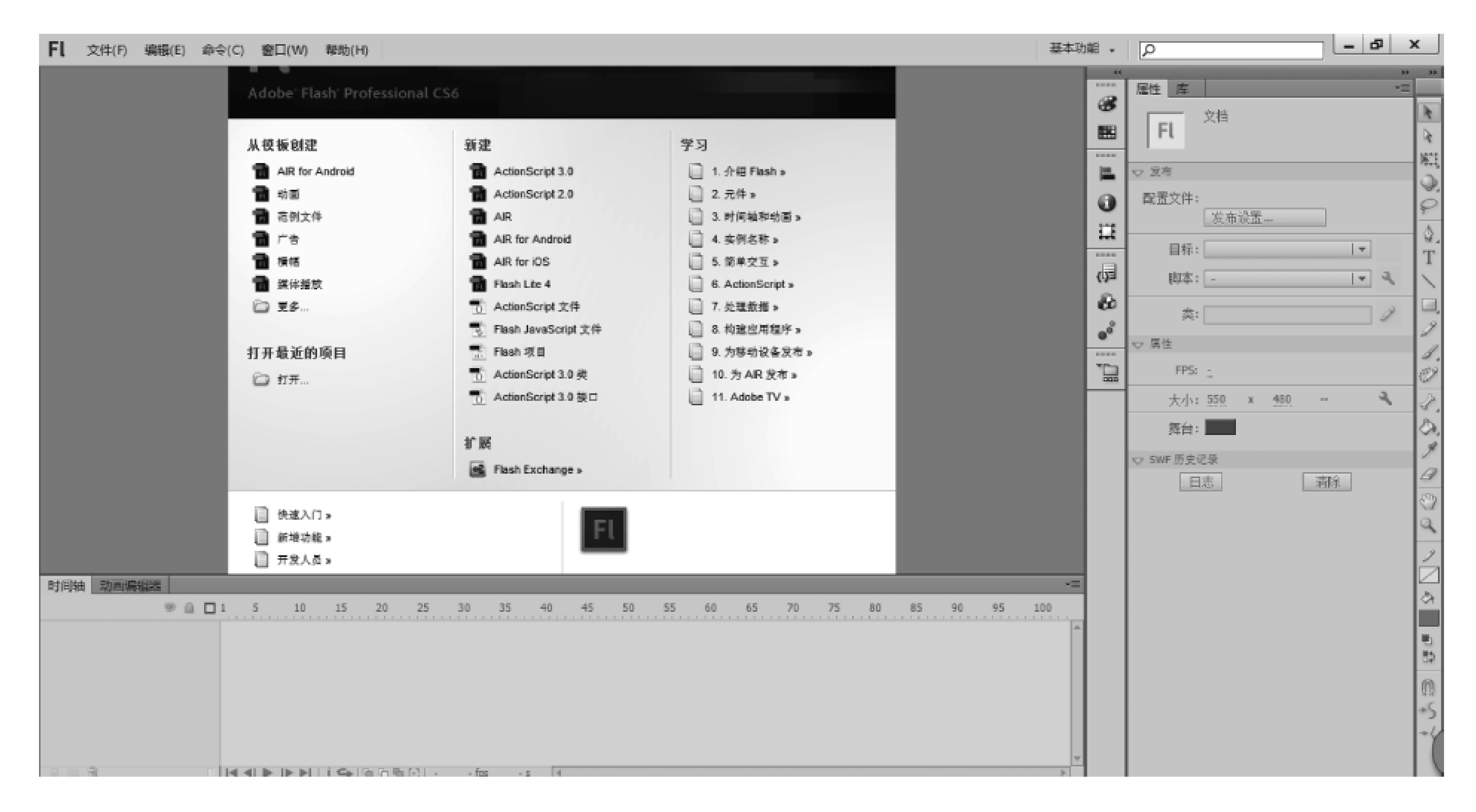


图 6-1 标准界面



图 6-2 【常规】选项卡参数

(2) 单击【新建】|ActionScript 3.0 项目，打开 Flash 工作界面，如图 6-3 所示。

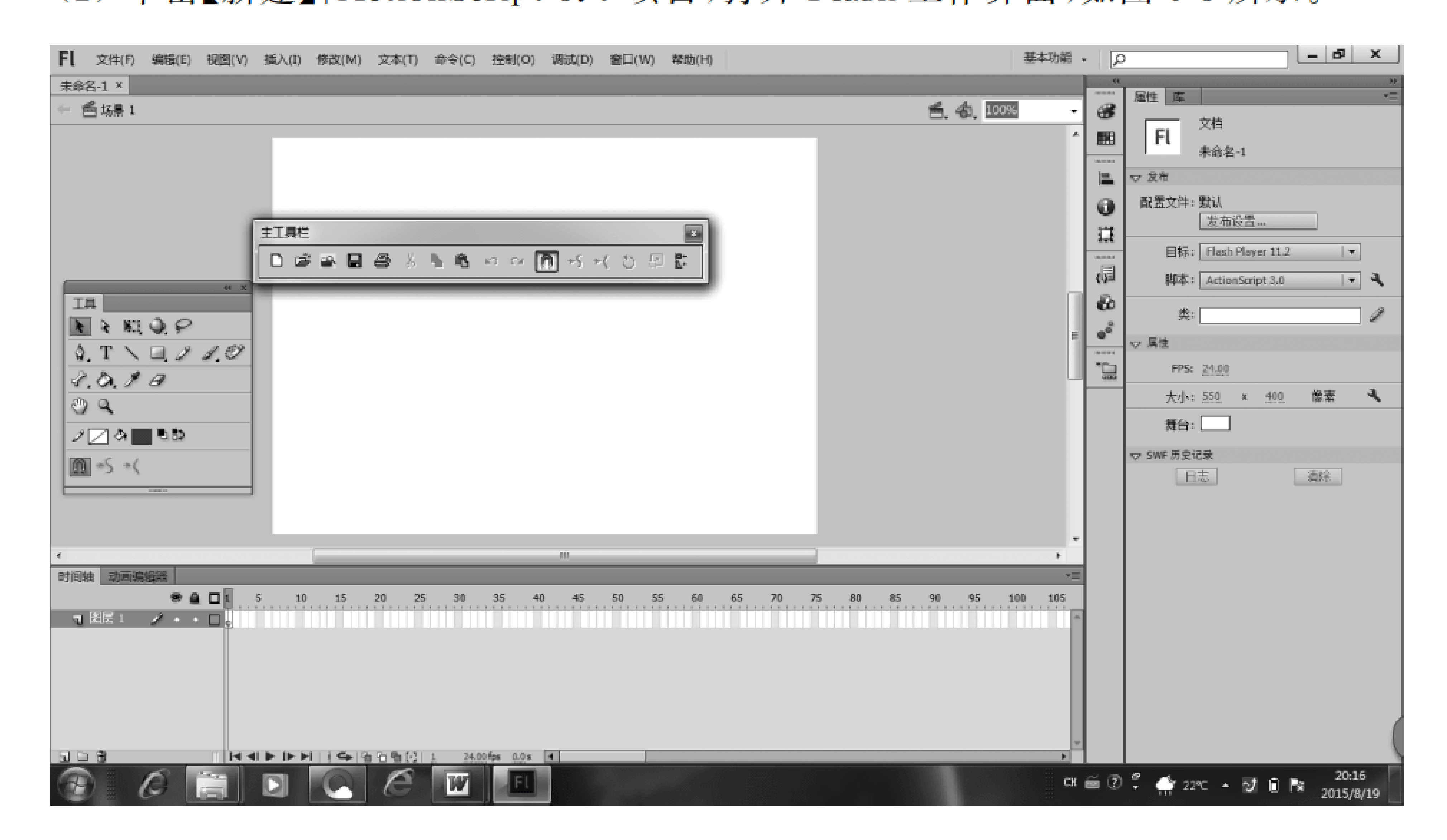


图 6-3　Flash 工作界面

该界面主要被划分成 7 个区域。

- 菜单栏：菜单栏是 Flash 界面的控制中心，可控制 Flash 文件和自定义工作环境。它包括了 11 个主菜单，每个主菜单的下拉菜单中还包含各种相应的操作命令和选项。为了方便用户操作，各主菜单下的很多子菜单都附有相应的快捷键，通过使用这些快捷键，用户可以直接执行相应的操作命令，从而提高效率。
- 主工具栏：将一些常用命令以按钮形式组织在一起，置于操作界面的上方。
- 工具箱：工具箱中提供了绘制和编辑图形的各种工具，分为"工具"、"查看"、"颜色"、"选项"4 个功能区。
- 时间轴：控制不同层中动画的帧数与帧的效果，包括新层的增加、引导层的建立、层的删除及相关帧的操作。
- 场景和舞台：场景是所有动画元素最大的活动空间，舞台是播放 Flash 电影时用户能看得到的区域。它包含所有可视的文档元素以及某些情况下不可视的文档元素，如文本、图像、视频和声音。
- 【属性】面板：用来修改文档和选定对象的面板。
- 浮动面板：可以快速激活各个常用面板，其中包括【属性】面板、【动作】面板、【颜色】面板等。

6.3 Flash CS6 的基本操作

对文件进行建立和保存是 Flash 中最基本的操作，同时文件的导入/导出是 Flash 中比较特殊的操作，大家要尤为重视。

6.3.1 文件的新建

(1) 启动 Flash CS6,新建一个 Flash 文档。选择【开始】菜单中的【程序】|Adobe Flash CS6 Professional 命令,启动 Flash 软件,出现图 6-1 所示的开始页,然后单击【新建】|ActionScript 3.0 项目,进入 Flash 工作界面,如图 6-3 所示。

(2) 在 Flash 工作界面中选择【文件】|【新建】命令,切换到【常规】选项卡,在【类型】中选择某一项,可新建文档。

6.3.2 文件的保存

当第一次保存文件时,【保存】命令与【另存为】命令等效,都是打开【另存为】对话框,可以在其中输入【文件名】和选择【类型】,保存文档。

当文件不是第一次保存时,【文件】菜单中的【保存】命令可以将文件覆盖保存;而【文件】菜单中的【另存为】命令则打开【另存为】对话框,选择不同的【文件名】或【类型】,可以将文档再存一份。

在 Flash 中,默认文件的保存格式为.fla,它是 Flash 的源文件,可以在 Flash 软件中打开、编辑和保存,Flash 中所有的原始素材和全部原始信息都保存在.fla 文件中。

6.3.3 设置影片属性

影片的【属性】面板默认出现在起始界面的右侧,也可以按 Ctrl+F3 键打开【属性】面板。【属性】面板用于显示或更改当前动画文档、文本、元件、形状、位图、视频、帧或工具的相关信息和设置,根据所选的对象不同,该面板中显示的内容也不同。如图 6-4 所示为文档的基本属性,给出了该文档的文件名、目标、脚本、大小、帧频等信息。



图 6-4 【属性】面板

6.3.4 设置首选参数

选择菜单栏中的【编辑】|【首选参数】命令，打开【首选参数】对话框，如图 6-5 所示。



图 6-5 【首选参数】对话框

在此对话框中可以对文档做初步设置，如启动界面、工作区选项、自动套用格式等。

6.3.5 文件的导入与导出

1. 文件的导入

选择菜单栏中的【文件】|【导入】命令，可将图像、声音、动画等文件导入到相应位置，如图 6-6 所示。

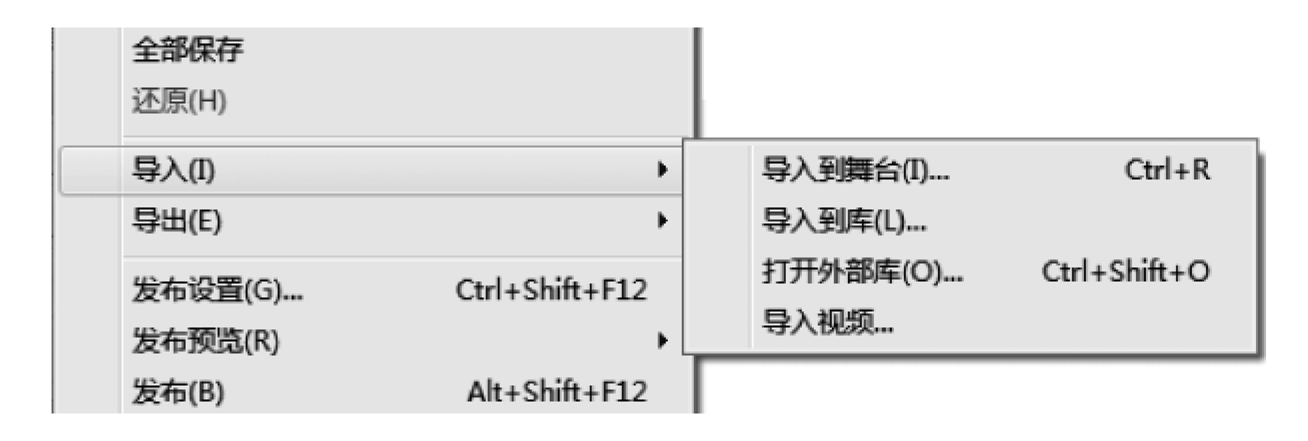


图 6-6 【导入】菜单

2. 文件的导出

选择菜单栏中的【文件】|【导出】命令,可将该文档导出为图像、动画、音频、图片序列等到相应位置,如图 6-7 所示。

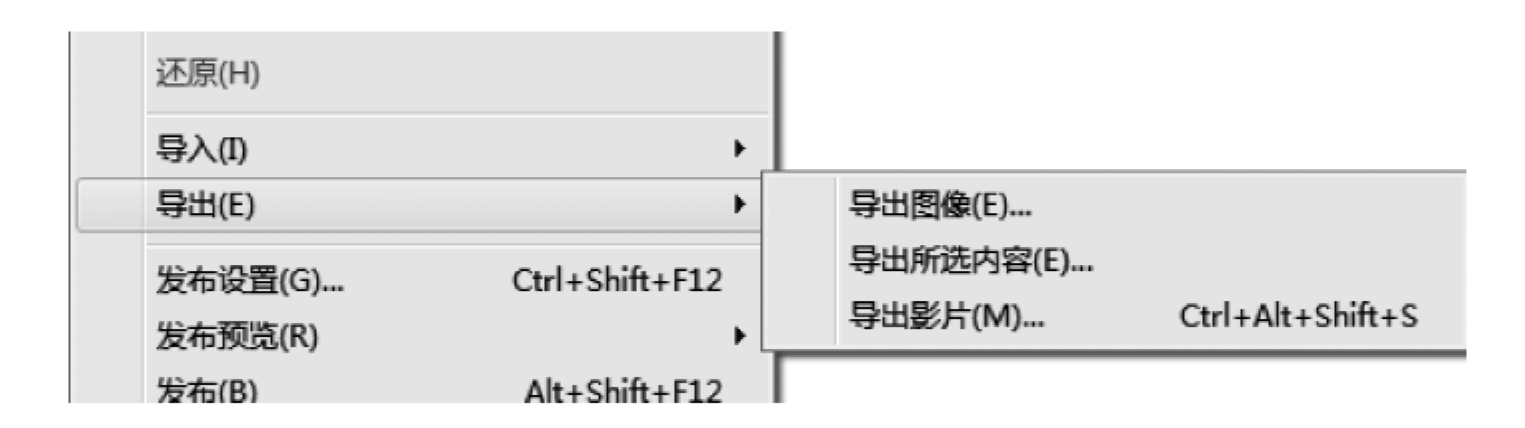


图 6-7 【导出】菜单

- 【导出图像】:可以将当前帧或所选图像导出为一种静止图像格式或导出为单帧 Flash Player 应用程序。
- 【导出所选内容】:可以将当前所选内容导出为一个扩展名为.fxg 的文件。
- 【导出影片】:可以将动画导出为包含一系列图片、音频的动画格式或静止帧。当导出静止图像时,可以为文档中的每一帧都创建一个带有编号的图像。另外,还可以将文档中的声音导出为 WAV 文件。

3. 输出影片格式

Flash CS6 可以输出多种格式的动画或图形文件,一般包含以下几种常见类型:

1) SWF 影片(*.swf)

SWF 动画是浏览网页时常见的动画格式,是以.swf 为扩展名的文件,具有动画、声音和交互等功能,它需要在浏览器中安装 Flash 播放器插件才能观看,将整个文档导出为具有动画效果和交互功能的 SWF 文件,以便将 Flash 内容应用到其他应用程序中,例如 Dreamweaver。

2) Windows AVI(*.avi)

Windows AVI 是标准的 Windows 影片格式,它是一种很好的用于在视频编辑应用程序中打开的 Flash 动画格式,由于 AVI 是基于位图的格式,因此如果包含的动画很长或者分辨率比较高,文件量会非常大。在将 Flash 文件导出为 Windows 视频时会丢失所有的交互功能。

3) WAV 音频(*.wav)

用户可以将动画中的音频对象导出,并以 WAV 声音文件格式保存。

4) JPEG 图像(*.jpg)

用户还可以将 Flash 文档中当前帧上的对象导出成 JPEG 位图文件,JPEG 格式图像为高压缩比的 24 位位图,JPEG 格式适合显示包含连续色调(如照片、嵌入位图或渐变色)的图像,其导出设置与位图(BMP)相似。

5) GIF 序列(*.gif)

网页中常见的动态图标大部分是 GIF 动画形式,它由多个连续的 GIF 图像组成,在

Flash 动画时间轴上的每一帧都会变为 GIF 动画中的一幅图片，GIF 动画不支持声音和交互，并且比不含声音的 SWF 动画文件量大。

6）PNG 序列（*.png）

PNG 文件格式是一种可以跨平台、支持透明度的图像格式。

6.4 Flash 动画的制作流程

制作 Flash 动画最基本的过程如下：

（1）设定舞台尺寸和安排场景。

（2）创建和插入动画成员。

（3）设定动画效果。

（4）预览并测试动画。

（5）保存、输出与发布动画。

6.5 综合应用

6.5.1 设置舞台的显示比率

选择菜单栏中的【视图】|【缩放比率】|【符合窗口大小】命令，如图 6-8 所示，可以让整个舞台显示在窗口内。



图 6-8 设置舞台的显示比率

6.5.2 设置显示网格线

选择菜单栏中的【视图】|【网格】|【显示网格】命令，该命令前若有“√”表示显示，如图 6-9 所示，再次选择可以隐藏。



图 6-9　显示网格

6.5.3　自定义工具面板

选择菜单栏中的【编辑】|【自定义工具面板】命令，打开【自定义工具面板】对话框，如图 6-10 所示。单击最左侧的工具按钮，然后将可用工具选中，单击【增加】按钮，就可以将该工具添加到在最左侧选中的工具按钮的隐藏菜单中，如图 6-11 所示。







图 6-10　添加工具　　　　图 6-11　添加工具后的效果

第7章 Flash CS6 工具箱

本章学习目标：

✍了解 Flash 工具箱中的所有工具。

✍掌握 Flash 工具箱中常用工具的使用。

7.1 绘制图形的工具

在制作 Flash 动画时，很多图形需要用户自己去绘制，Flash 自带了一些绘图工具，用户使用这些工具可以绘制出丰富多样的图形。Flash 中的绘图工具包括线条工具、铅笔工具、椭圆工具、矩形工具、刷子工具和钢笔工具等。

7.1.1 线条工具

线条工具主要用于绘制不同角度的矢量直线。选择线条工具，在舞台上单击鼠标，按住鼠标左键不放并向右拖动到需要的位置，松开鼠标，即可绘制出一条直线，效果如图 7-1 所示。在线条工具的【属性】面板中可以设置笔触颜色、笔触大小、笔触样式，如图 7-2 所示。

图 7-1　直线



图 7-2　线条工具的【属性】面板

选择线条工具后，在工具箱下面会出现【对象绘制】和【贴紧至对象】两个按钮。

【对象绘制】：当选择了【对象绘制】按钮时，在舞台中绘制的“直线”将自动转换为“组”。

【贴紧至对象】：当选择了【贴紧至对象】按钮时，在绘制对象时，线段会开启“吸附”功

能，可以更好地绘制对象。在图中，当一条线接近另一条线时，在线段的一端会出现一个“空心圆”，它可以很自然地和另一条线段连接在一起。在绘制对象时，经常会使用【贴紧至对象】按钮，用户可以根据具体情况对这个功能进行切换。

- 【笔触颜色】：设置线段的颜色。
- 【笔触高度】：设置线段的高度。
- 【笔触样式】：在该下拉列表中可以选择各种线段样式。
- 【提示】：将笔触锚点保持为全像素可防止出现模糊线。
- 【端点】：可以选择“圆角”或“方形”笔触。

7.1.2 铅笔工具

铅笔工具主要利用前景色绘制任意线条。选择铅笔工具，在舞台上单击鼠标，按住鼠标左键不放，松开鼠标，在舞台上随意绘制线条，效果如图 7-3 所示。如果想要绘制出平滑或伸直的线条和形状，可以在工具箱下方的选项区中为铅笔工具选择一种绘画模式，如图 7-4 所示。

- 【伸直】：可以绘制直线，并将接近三角形、椭圆、圆形、矩形和正方形的形状转换为这些常见的几何形状。
- 【平滑】：可以绘制平滑曲线。
- 【墨水】：可以绘制接近手绘的线条。

在铅笔工具的【属性】面板中可以设置不同的笔触颜色、笔触大小、笔触样式，如图 7-5 所示。

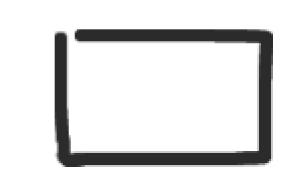


图 7-3 铅笔绘制效果



图 7-4 铅笔的绘画模式



图 7-5 铅笔工具的【属性】面板

7.1.3 椭圆工具

椭圆工具主要用于绘制椭圆图形，它和矩形工具类似，差别在于椭圆工具的选项中有关于角度和内径的设置。选择椭圆工具，在舞台上单击，按住鼠标左键不放向需要的位置拖动，松开鼠标，即可绘制出椭圆，效果如图 7-6 所示。在椭圆工具的【属性】面板中可以设置

不同的笔触颜色、笔触大小、笔触样式，如图 7-7 所示。

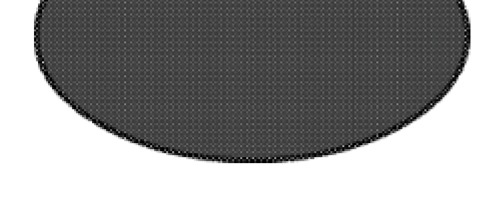


图 7-6　椭圆



图 7-7　椭圆工具的【属性】面板

- 【开始角度】：设置要绘制圆的起始角度。
- 【结束角度】：设置要绘制圆的结束角度，和【开始角度】一起使用会绘制一个饼图。
- 【闭合路径】：勾选该复选框会绘制一个圆环。

7.1.4　矩形工具

矩形工具主要用于绘制矩形图形，这些图形不仅能够设置形状、大小、颜色，还可以设置边角半径以修改矩形形状。选择矩形工具，在舞台上单击鼠标，按住鼠标左键不放向需要的位置拖动，松开鼠标，即可绘制出矩形图形，效果如图 7-8 所示。在矩形工具的【属性】面板中可以设置不同的笔触颜色、笔触大小、笔触样式和填充颜色，如图 7-9 所示。

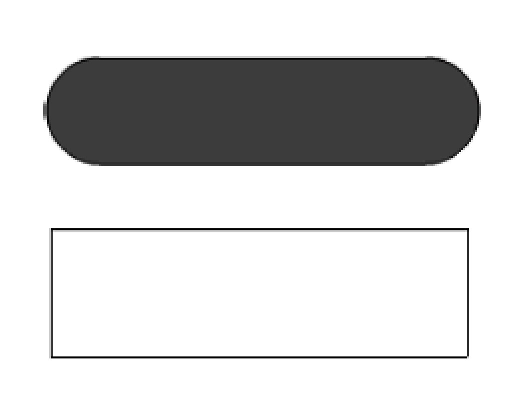


图 7-8　矩形



图 7-9　矩形工具的【属性】面板

与其他图形工具不同的是矩形选项部分可以设置矩形的圆角，这样绘制出来的就是圆角矩形。

7.1.5 多角星形工具

多角星形工具主要用于绘制多边形和多角星形，使用多角星形工具可以绘制出不同样式的多边形和星形。选择多角星形工具，在舞台上单击，按住鼠标左键不放向需要的位置拖动，松开鼠标，即可绘制出多边形，效果如图 7-10 所示。在多角星形工具的【属性】面板中可以设置不同的笔触颜色、笔触大小、笔触样式和填充颜色，如图 7-11 所示。



图 7-10 多角星形



图 7-11 多角星形工具的【属性】面板

单击【属性】面板下方的【选项】按钮，弹出【工具设置】对话框，在该对话框中可以自定义多边形的各种属性。

- 【样式】选项：在此选择绘制多边形或星形。
- 【边数】选项：设置多边形的边数，其范围为 3～32。
- 【星形顶点大小】选项：输入一个 0～1 之间的数字以指定星形顶点的深度，此数字越接近 0，创建的顶点就越深，此选项在多边形的绘制中不起作用。

在设置不同值以后，绘制出的多边形和星形不同。

7.1.6 刷子工具

选择刷子工具，在舞台上单击鼠标，按住鼠标左键不放拖动，随意绘制，松开鼠标，效果如图 7-12 所示。刷子工具的【属性】面板与其他绘图工具的【属性】面板相似，可以在其中设置不同的填充颜色和笔触平滑度等，但是在选择刷子工具后，在工具箱下方会出现【刷子模式】按钮，如图 7-13 所示。



图 7-12 刷子绘制效果　　图 7-13 刷子模式

系统在工具箱的下方提供了 5 种刷子的模式。

- 【标准绘画】模式：在同一层的线条和填充上以覆盖的方式涂色。
- 【颜料填充】模式：对填充区域和空白区域涂色，其他部分(如边框线)不受影响。
- 【后面绘画】模式：对舞台上同一层的空白区域涂色，但不影响原有的线条和填充。
- 【颜料选择】模式：在选定的区域内涂色，未被选中的区域不能涂色。
- 【内部绘画】模式：在内部填充上绘图，但不影响线条，如果从空白区域中开始涂色，该填充不会影响任何现在填充区域。

7.1.7 喷涂刷工具

喷涂刷工具类似于粒子喷射器，使用它可以一次将形状图案刷到舞台上，如果使用喷涂刷工具，可在工具栏中按住常规刷工具，直到弹出【刷选择】菜单，再单击【喷涂刷】。与使用常规刷一样在舞台上绘制，喷出的一些圆形小颗粒将显示为一组。单击【属性】面板中的拾色器并选择另一种颜色，更改刷子的颜色，然后在舞台上绘制其他内容，如图 7-14 所示。

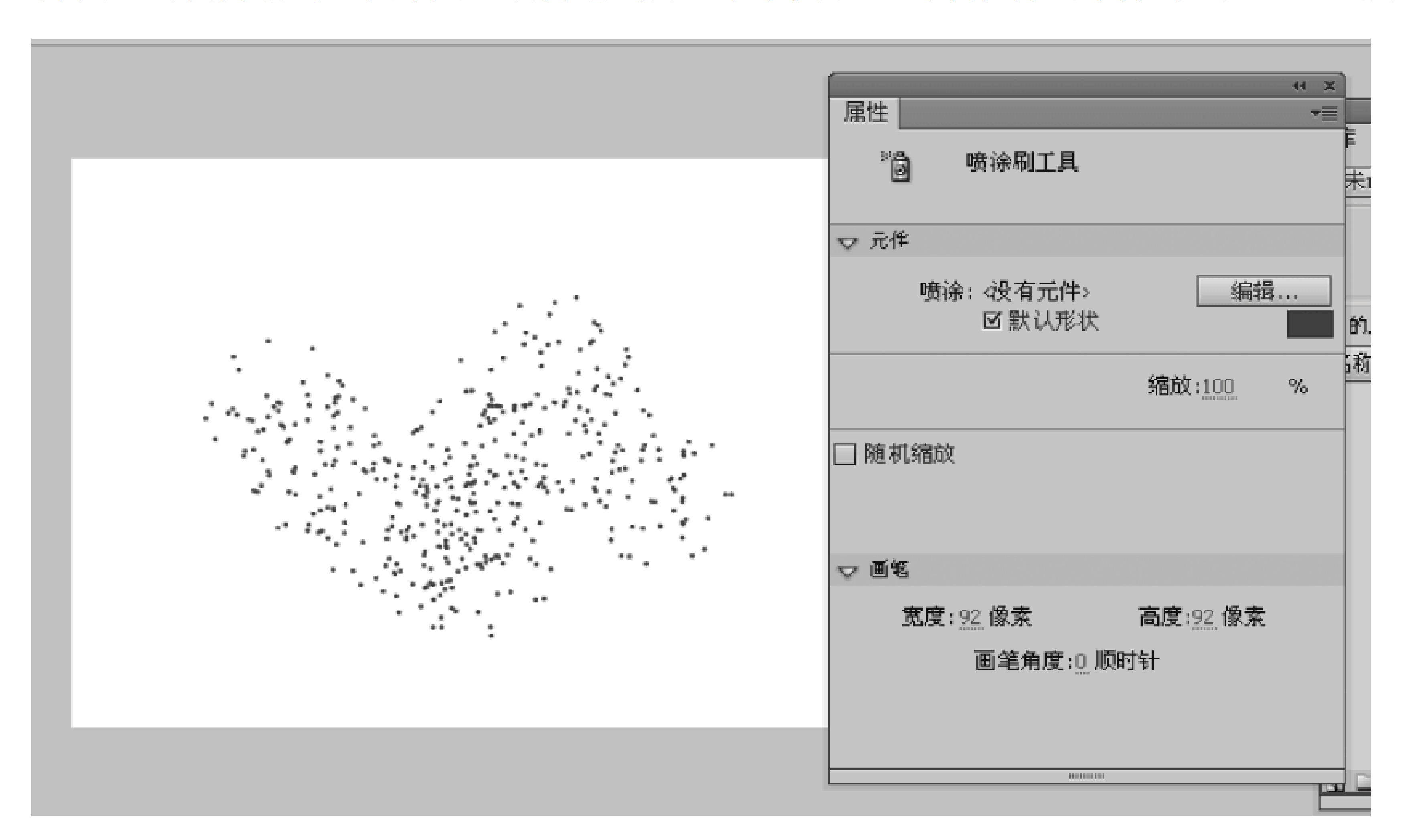


图 7-14 喷涂刷工具效果及【属性】面板

- 【元件】:可以看到喷涂刷工具的【属性】面板默认勾选【默认形状】复选框,用户可以建立自己的图形,然后选择图形并按 F8 键转换为图形元件或影片剪辑元件,这样在喷涂时就可以选择用户刚准备好的图形元件或影片剪辑元件了。在选择了元件之后,就可以进行旋转元件或随机旋转的操作了。
- 【缩放】:调整喷涂刷工具的喷涂大小,可以设置为 0～40000 之间的数值。
- 【随机缩放】:勾选后所喷涂的图形为随机大小,这样更符合喷涂刷工具的实际使用效果。
- 【画笔】:在列表中可以对【宽度】、【高度】和【画笔角度】进行设置。【宽度】和【高度】分别对应喷涂刷工具的喷涂范围,如将【宽度】设置为 0 像素,将【高度】设置为 50 像素,那么喷涂出来的图形就是一个竖条图形。【画笔角度】可以对喷涂的基础图形进行角度的设置,这样可以模仿实际情况下的喷涂方向。

7.1.8 钢笔工具

钢笔工具主要用于绘制比较复杂、精确的曲线路径。选择钢笔工具,将鼠标指针放置到舞台上想要绘制曲线的起始位置,然后按住鼠标左键不放,此时将出现第一个锚点,并且钢笔尖光标变为箭头形状,松开鼠标,将鼠标指针放置到想要绘制的第二个锚点的位置,单击鼠标并按住鼠标左键不放绘制出一条直线段,将鼠标向其他方向拖动,直线转换为曲线,松开鼠标,一条曲线绘制完成,如图 7-15 所示。



图 7-15 用钢笔工具绘制曲线

在绘制线段时钢笔工具的光标会产生不同的变化,其表示的含义不同。

- 增加节点:当光标变为带加号时,在线段上单击鼠标就会增加一个节点,这样有助于更精确地调整线段。
- 删除节点:当光标变为带减号时,在线段上单击一个节点,就会将这个节点删除。
- 转换节点:当光标变为带折线时,在线段上单击节点,就会将这个节点从曲线节点转换为直线节点。

7.1.9 Deco 工具

Deco 工具是装饰性绘画工具,可以将创建的图形形状转变为复杂的几何图案。首先制作影片剪辑元件,在【绘制效果】中选择某一效果,单击【编辑】按钮,将所做元件与动画联系到一起,在【高级】中进行相应的属性设置后,在舞台上单击,自动生成动画。其中,由于选择的绘制效果不同,其下方出现的参数也不同,有的出现两个元件需要编辑,有的不需要元件。Deco 工具的【属性】面板如图 7-16 所示。

当绘制效果为藤蔓式填充时参数如下。

- 【树叶】与【花】:设置的是将已有影片剪辑进行编辑。
- 【分支角度】:指定分支图案的角度。
- 【图案缩放】:缩放操作会使对象同时沿着水平方向和垂直方向放大或者缩小。



图 7-16　Deco 工具的【属性】面板

- 【段长度】：指定叶子节点和花朵节点之间的段长度。
- 【动画图案】：指定效果的每次迭代都绘制到时间轴中的新帧。在绘制花朵图案时，勾选此复选框将创建花朵图案的逐帧动画序列。
- 【帧步骤】：指定绘制效果时每秒要横跨的帧数。

7.2　文本工具

文本是任何动画都不可或缺的元素之一，在 Flash CS6 中提供了方便的文本输入与编辑功能。

7.2.1　文本的输入

(1) 选择工具箱中的文本工具，将鼠标指针移动到舞台上，单击鼠标，开始输入。

(2) 选择工具箱中的文本工具，将鼠标指针移动到舞台上，按住鼠标左键拖动，出现一个矩形文本框，代表此文本输入框有宽度限制，当输入的字符超过输入框的宽度时，文字会自动换行。

7.2.2　文本的类型

在 Flash 中输入的文本可以分为 3 种类型，分别是静态文本、动态文本和输入文本，这 3 种文本类型可以在文本工具的【属性】面板的文本类型下拉列表框中选择。

(1) 静态文本：一种静止不动的文本，是在 Flash 动画运行过程中不能修改的文字。

（2）动态文本：在 Flash 动画运行过程中可以动态显示的文字，通常需要结合 AS 动态显示一些数据或运算结果。

（3）输入文本：在 Flash 动画运行过程中可以输入文字，与动态文本一样，需要结合 AS 达到动画和用户交互的目的。

7.2.3　文本工具的【属性】面板

当选择文本工具后，或者在舞台上输入文本后，打开【属性】面板，可以设置文本的相关属性，如图 7-17 所示。



图 7-17　文本工具的【属性】面板

- 【文本引擎】：在 Flash CS6 中，【文本引擎】有 TLF 和传统文本两种，其中 TLF 文本是一种文本引擎——文本布局框架向 Flash 中添加文本，它可以支持更多、更丰富的文本布局功能和对文本属性的精细控制。
- 【文本类型】：共有 3 种，分别是静态文本、动态文本和输入文本。
- 【系列】：设置选定字符或整个文本的文字字体。
- 【样式】：设置字体的样式。
- 【嵌入】：所选中字体及样式是否嵌入文档中。
- 【大小】：用于设置文字字体的大小。
- 【字母间距】：设置文本输入框中每个字符之间的间距。
- 【颜色】：用于设置文字的预色，单击【颜色】按钮会弹出一个颜色设置面板，可以从中选择需要的文字颜色。
- 【左对齐】：设置文本框中的文字左对齐。
- 【居中对齐】：设置文本框中的文字居中对齐。
- 【右对齐】：设置文本框中的文字右对齐。
- 【两端对齐】：设置文本框中的文字两端对齐。
- 【间距】：设置文字的首行缩进与行间距。
- 【边距】：设置文字的左右缩进。

7.3　图形编辑工具

利用图形编辑工具可以对图形进行编辑和制作，从而节省制作动画的工作量。在绘制图形时合理地运用编辑工具可以使绘制的图形更加丰富多彩，达到事半功倍的效果。在 Flash CS6 中，编辑图形的工具包括颜料桶工具、墨水瓶工具、滴管工具、橡皮擦工具、渐变变形工具和任意变形工具。

7.3.1　颜料桶工具

颜料桶工具可用于为封闭或者半封闭的图形区域填充颜色，它可以为图形填充单色，也

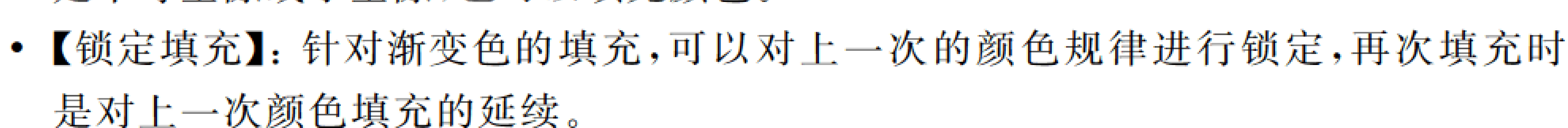
可以为图形填充渐变颜色，还可以为图形填充位图图形，但是不能为图形的线段填充颜色。

选择颜料桶工具，在工具箱下方会出现两个按钮，如图 7-18 所示。在工具箱下方设置了 4 种填充模式供用户选择。



图 7-18 颜料桶工具的填充模式

- 【不封闭空隙】模式：选择此模式，只有在完全封闭的区域颜色才能被填充。
- 【封闭小空隙】模式：选择此模式，当边线上存在小空隙时允许填充颜色。
- 【封闭中等空隙】模式：选择此模式，当边线上存在中等空隙时允许填充颜色。
- 【封闭大空隙】模式：选择此模式，当边线上存在大空隙时允许填充颜色。当选择【封闭大空隙】模式时，如果空隙是中等空隙或小空隙，也可以填充颜色。



- 【锁定填充】：针对渐变色的填充，可以对上一次的颜色规律进行锁定，再次填充时是对上一次颜色填充的延续。

颜料桶工具的【属性】面板与线条工具的【属性】面板相似，此处不再叙述。

7.3.2 墨水瓶工具

墨水瓶工具主要用来改变已有直线或曲线的颜色、粗细和线型等属性，此外还可以为没有边框的图形添加轮廓线。

墨水瓶工具的【属性】面板与线条工具的【属性】面板相似，此处不再叙述。

7.3.3 滴管工具

滴管工具主要用于从线段、图形和文本中吸取填充颜色、笔触属性和文字属性，然后将它们应用到其他对象上。此外，滴管工具还允许从位图上进行采样作为对象的填充内容。

7.3.4 橡皮擦工具

使用橡皮擦工具可以擦除图形的填充颜色和线条，擦除打散的位图，还可以一次性擦除图形的填充颜色和线条。

当选择橡皮擦工具后，在工具箱中会出现【橡皮擦模式】、【水龙头】、【橡皮擦形状】3 个按钮，如图 7-19 所示。

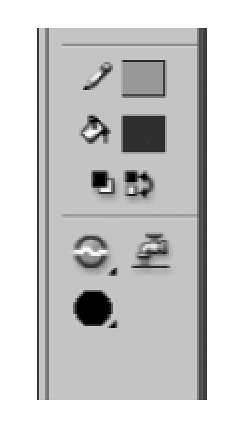


图 7-19 橡皮擦工具选项

1. 【橡皮擦模式】按钮

【橡皮擦模式】按钮主要用来设置橡皮擦工具的不同擦除模式，每种模式都有不同的擦除效果，在单击【橡皮擦模式】按钮后有以下几种模式。

- 【标准擦除】：选择此模式，以常规模式擦除图形，可以把鼠标经过地方的图形全部擦除。

- 【擦除填色】：选择此模式，可以把鼠标经过地方的填充颜色擦除，不能擦除线条。
- 【擦除线条】：选择此模式，可以把鼠标经过地方的线条擦除，不能擦除填充颜色。
- 【擦除所选填充】：选择此模式，只能擦除选择范围内的填充颜色，不能擦除选择范围内的线条。
- 【内部擦除】：选择此模式，只能擦除鼠标经过的第一个填充区域内的颜色，对其他区域没有影响。

2.【水龙头】按钮

【水龙头】按钮主要用于将图形的填充色或者线条一次性擦除。

3.【橡皮擦形状】按钮

在其下拉列表中可以设置橡皮擦的大小和形状。

7.3.5 渐变变形工具

在为图形填充渐变颜色或者位图后，可以使用渐变变形工具进一步编辑图形填充颜色或者位图图形。它可以调整渐变颜色和位图的范围、方向、角度等，从而使图形的填充效果更符合要求。使用渐变变形工具可以调整线性渐变、放射状渐变和位图填充。如图 7-20 所示拖动各个圆点，将对渐变颜色进行调整。

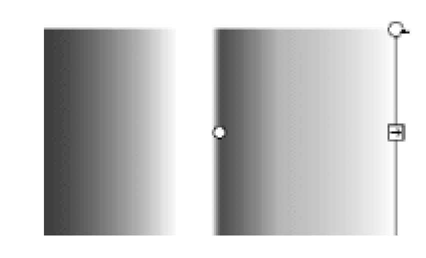


图 7-20 渐变变形工具效果

7.3.6 任意变形工具

任意变形工具在编辑图形时经常用到，它通常用于改变对象的大小及旋转与倾斜对象。在使用任意变形工具选择要编辑的对象后，舞台上对象四周将出现 8 个控制点，如图 7-21 所示。可以对图像进行变形操作，在工具箱中会出现【旋转与倾斜】、【缩放】、【扭曲】、【封套】4 个按钮，如图 7-22 所示。

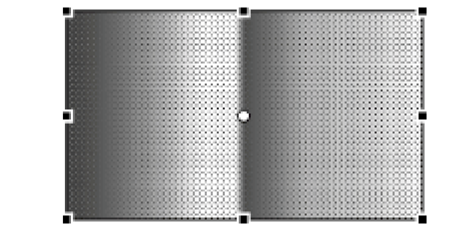


图 7-21 任意变形工具效果



图 7-22 任意变形工具选项

需要注意的是,【扭曲】和【封套】两个按钮只有在所选对象是打散的图形或者是没有成组的矢量图时才能应用。如果没有选中任何一个按钮,任意变形工具处于自由变形模式,除了【封套】功能外,其他的操作都可以进行。

- 【紧贴至对象】:使选择的图像吸附于舞台的网格、辅助线。
- 【旋转与倾斜】:此工具可以将选中的对象进行任意旋转,以及水平或垂直方向上的倾斜变形。
- 【缩放】:此工具可以将选中的对象做水平、垂直缩放或等比例缩放。
- 【扭曲】:此工具可以通过调节选取对象四周的节点位置改变图形的形状。
- 【封套】:可以通过调节选取对象的 4 条边线上的控制点或者控制点上的控制手柄来改变图形的形状。

7.3.7 骨骼工具和绑定工具

Flash 骨骼工具提供了对骨骼动力学的有力支持,利用骨骼工具可以实现多个符号或物体的动力学连动状态。创建骨骼动画有两种方式,一种是给元件实例添加,另一种是给图形元件添加。在此以图形为例,选中舞台上的图形,选择工具箱中的骨骼工具,使用骨骼工具在形状内单击并拖动到形状内的其他位置,在拖动时将显示骨骼,选择骨骼绑定工具,把鼠标指针放置到创建好的骨骼关节处,然后按住鼠标左键拖动,此时图形周围出现了小黄点,这些小黄点就是骨骼绑定工具要绑定的点,将鼠标指针拖动到这些点上松开,绑定就完成了,绑定后的骨骼将通过这个点影响图形形状,最后使用选择工具拖动骨骼查看动画效果,如图 7-23 所示。

图 7-24 所示为骨骼【属性】面板,在其中可以对骨骼进行坐标及宽、高的设置。

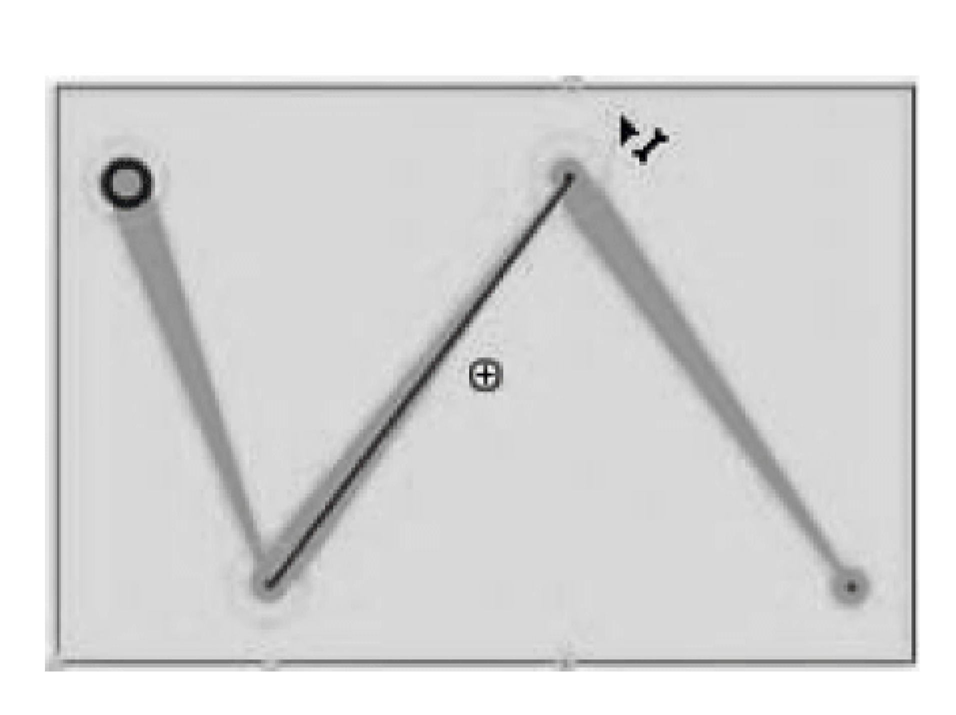


图 7-23 建立骨骼效果



图 7-24 骨骼【属性】面板

7.4 辅助工具

在 Flash CS6 中提供了选择工具、部分选取工具和套索工具，主要用于对对象进行选择、移动与编辑等操作。

7.4.1 选择工具

使用选择工具可以对图形进行选择、移动与编辑等操作，选择选择工具后，工具箱下方会出现其选项按钮，如图 7-25 所示。

- 【紧贴至对象】：该按钮的作用是使选择的图像吸附于舞台的网格、辅助线。
- 【平滑】：该按钮的作用是使所选择线条的弯曲处变得比较光滑。
- 【伸直】：该按钮的作用是使所选择线条的弯曲处变得比较尖锐。

图 7-25 选择工具选项

1. 选择对象

选择选择工具，在舞台上单击进行对象的选取，按住 Shift 键可选取多个对象。

2. 移动对象

选择对象后，按住鼠标左键将其拖动到新位置。

3. 复制对象

选择对象后，在按住 Alt 键的同时按住鼠标左键将选中对象拖动到新位置。

7.4.2 部分选取工具

部分选取工具通常和钢笔工具结合使用，主要用于调整图形的路径、编辑图形。

选择部分选取工具，在对象的外边线上单击，对象上会出现多个节点，通过拖动节点来调整控制线的长度和斜率，从而改变对象的曲线形状。

7.4.3 套索工具

套索工具主要用来选择位图图形的一部分。在使用套索工具之前必须将位图打散（快捷键是 Ctrl+B)，然后在图形上按住鼠标左键按照想要选择的图形轮廓拖动鼠标拖出一个封闭的区域，这样即可将该区域中的图形选中。



图 7-26 套索工具选项

在选择套索工具后，工具箱中会出现【魔术棒】、【魔术棒设置】和【多边形】3 个选项按钮，如图 7-26 所示。

- 【魔术棒】按钮：以点选的方式选择颜色相似的位图图形。
- 【魔术棒设置】按钮：用来设置魔术棒的属性。
- 【多边形】按钮：可以用鼠标精确地勾画想要选中的图像。

7.4.4 3D 旋转工具和 3D 平移工具

在 Flash CS6 中提供了 Z 轴的概念，也就是可以创建三维的动画效果，该工具仅对影片剪辑有效。

将某影片剪辑应用到舞台上，形成实例，然后选中该实例，选择 3D 旋转工具，选中的实例图形效果如图 7-27 所示，拖动光圈内的控制点，让动画沿某一轴旋转，即 3D 旋转。如果想让该动画在旋转过程中有位置移动，则选择 3D 平移工具拖动实例到新位置。

3D 旋转工具和 3D 平移工具的【属性】面板一致，如图 7-28 所示。

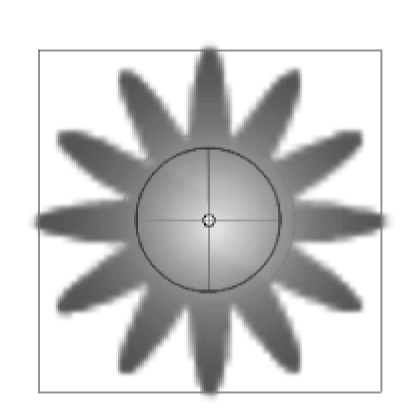


图 7-27　3D 旋转工具的应用



图 7-28　3D 旋转工具和 3D 平移工具的【属性】面板

7.5 综合应用

7.5.1 扇面的绘制

绘制扇面的操作步骤如下：

(1) 启动 Flash，选择菜单栏中的【文件】|【新建】命令，创建 800 像素×600 像素的 Flash 文档，其余参数默认，如图 7-29 所示。

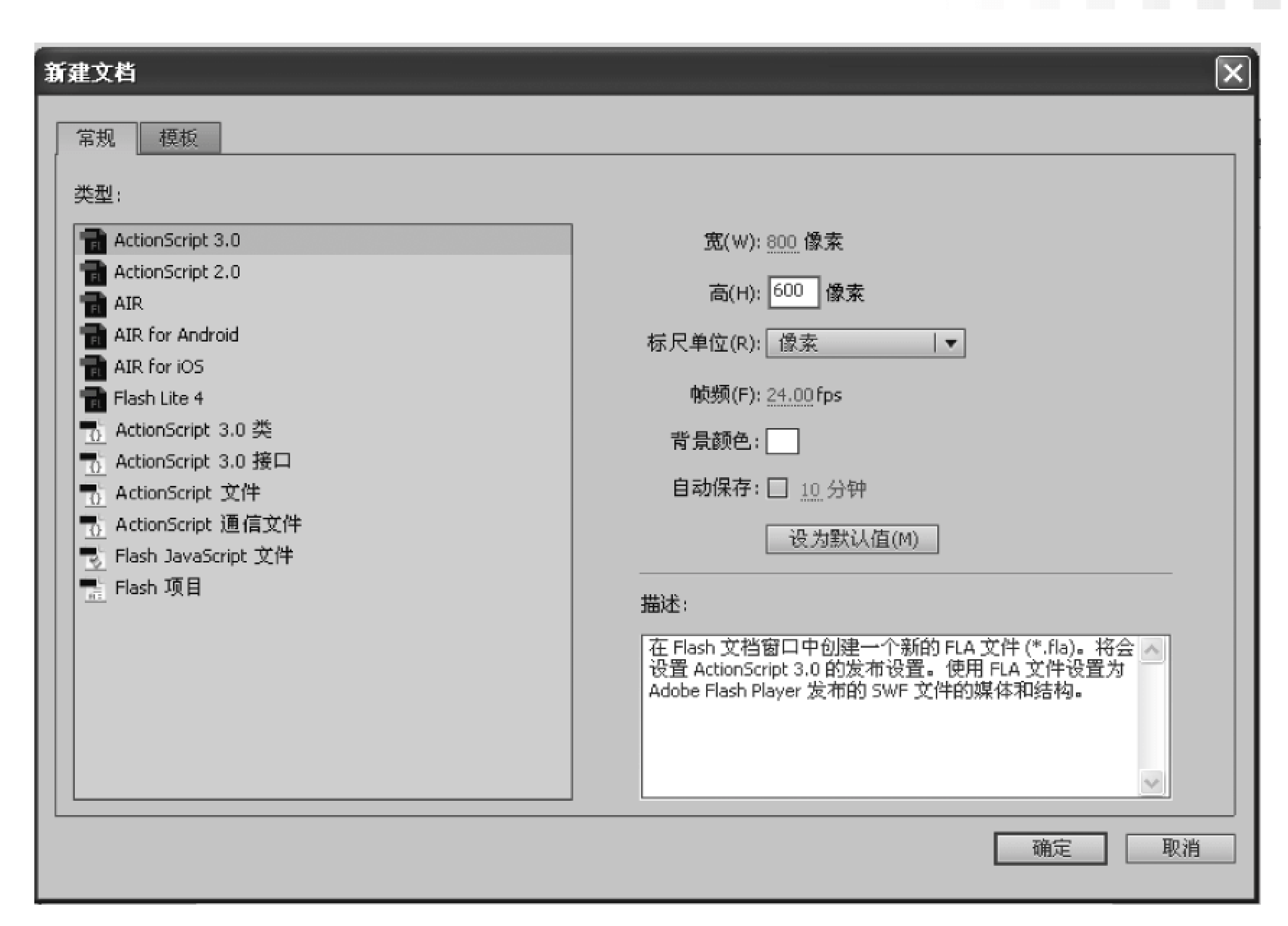


图 7-29 新建文档

(2) 选择图层 1，然后选择工具箱中的矩形工具，在舞台上拖动绘制长条矩形，并用油漆桶工具为其填充褐色，用部分选取工具选中矩形的右上角控制点向下拖动，同样，将右下角控制点向上拖动，将矩形调为左粗右细，如图 7-30 所示。

图 7-30 绘制并调整矩形

(3) 选中已修改的矩形，选择【修改】|【转换为元件】命令，将其转换为图形元件，如图 7-31 所示。



图 7-31 将矩形转换为图形元件

(4) 选中舞台上的元件实例，按 Ctrl+T 键打开【变形】面板，将矩形中心点向细的方向移动，并按图 7-32 所示将旋转角度改为 15°，单击【重制选区并变形】按钮，将矩形进行复制。

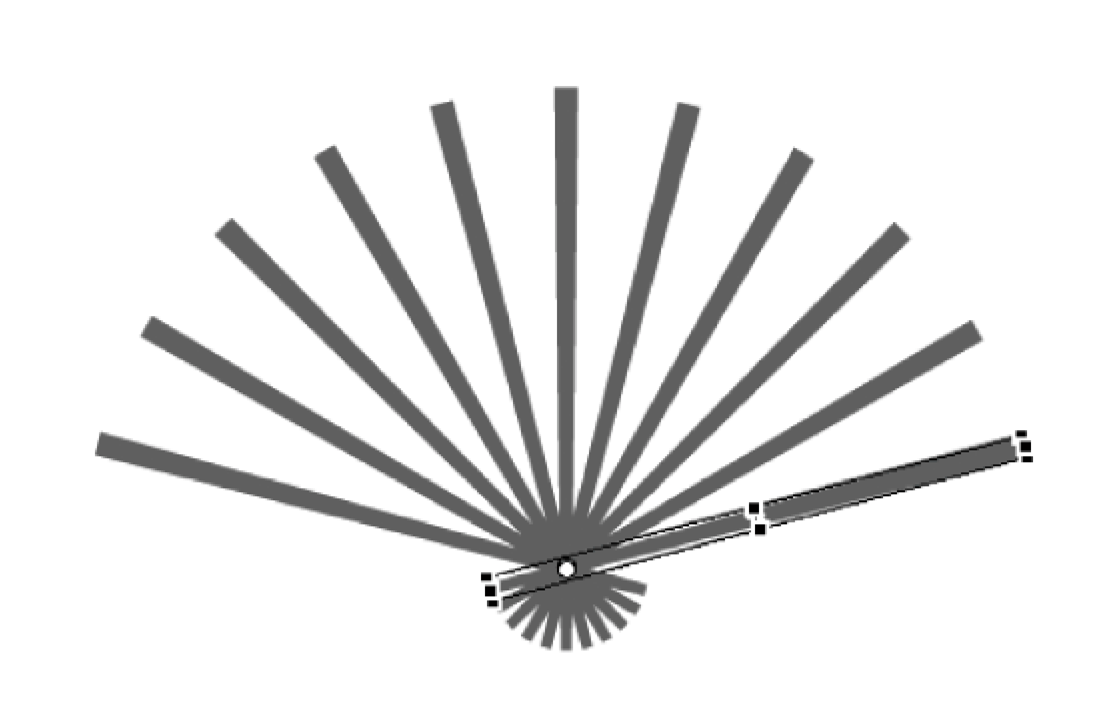




图 7-32　复制矩形

(5) 单击【图层】面板下方的【新建图层】按钮创建新层，在其中绘制小正圆，以黑白径向填充，放在扇子骨的交界处，如图 7-33 所示。

(6) 选择椭圆工具，以扇子的中心为中心，按 Shift+Alt 键从中心点开始绘制正圆，然后选中正圆，按 Ctrl+C 键将正圆复制，选择【编辑】|【粘贴到当前位置】命令，接着选择工具箱中的变形工具，按住 Shift+Alt 键，从中心成比例缩放，得到一个小圆，并用线条工具在扇子两侧绘制直线，删掉多余地方的线，得到一个半弧形，如图 7-34 所示。

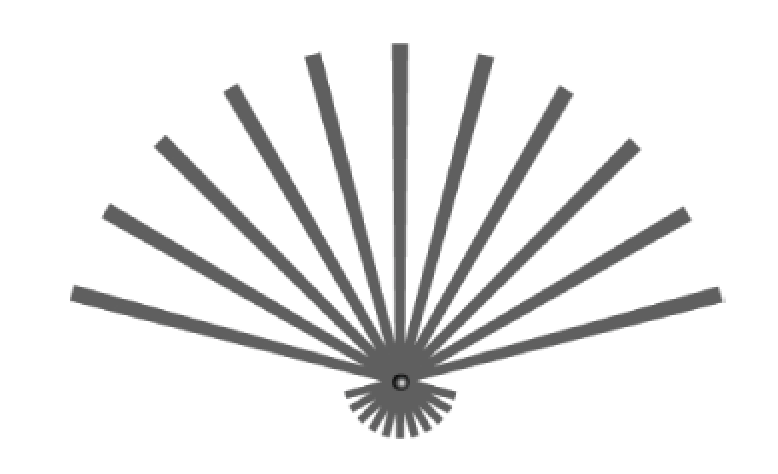


图 7-33　绘制正圆

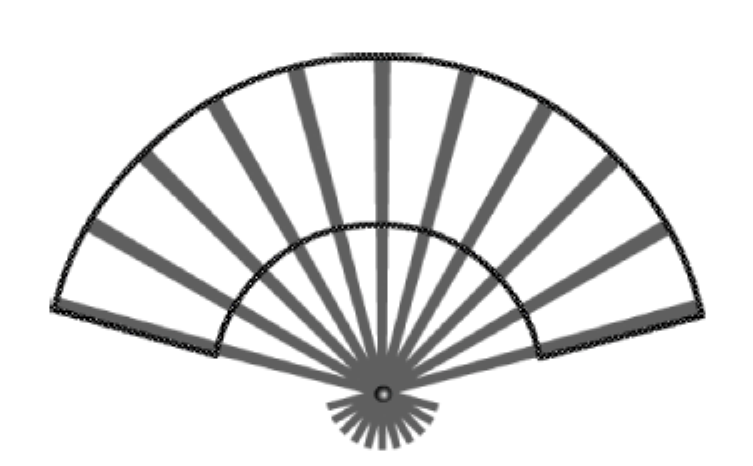


图 7-34　绘制扇面

(7) 选择【窗口】|【颜色】命令，在【颜色】面板上选择【位图填充】，将做扇面的图像导入，然后用油漆桶工具对扇形进行填充，如图 7-35 所示。

(8) 将扇面选中，按 F8 键将扇面转换成图形元件，将其透明度设为 90%。然后选中扇子右侧的最后一根扇骨，将它调到最上层。最后保存、预览，最终效果如图 7-36 所示。



图 7-35　填充扇面



图 7-36　扇子效果图

7.5.2　商品标识的制作

制作商品标识的操作步骤如下：

(1) 选择菜单栏中的【文件】|【新建】命令，创建420像素×420像素的Flash文件，其余参数默认。

(2) 选中图层1的第1帧，选择工具箱中的矩形工具，在舞台上绘制一个420像素×420像素的矩形。然后选择油漆桶工具，在【颜色】面板中选择以黄色到土黄色径向渐变填充，如图7-37所示。

(3) 单击【图层】面板下方的【新建图层】按钮，创建图层2，然后选择菜单栏中的【文件】|【导入】|【导入到舞台】命令，将"橙汁标识"素材导入到舞台，并调整大小、位置，如图7-38所示。

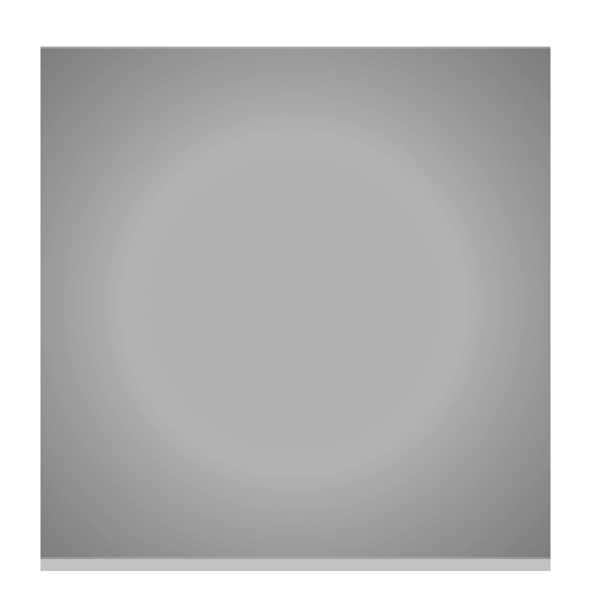


图7-37　绘制矩形并填充



图7-38　导入素材并调整

(4) 创建图层3，选择工具箱中的文本工具，在舞台中输入"鲜橙汁"3个字，在【属性】面板中将文字设置黑体、40像素、橙色，并按Ctrl+B键打散。然后选择墨水瓶工具，设置线条颜色为白色，在文字上单击，为文字添加白色边框，如图7-39所示。

(5) 选择文字，然后选择菜单栏中的【修改】|【变形】|【封套】命令，文字四周出现控制点，按住控制点进行拖动，将文字变形，如图7-40所示。

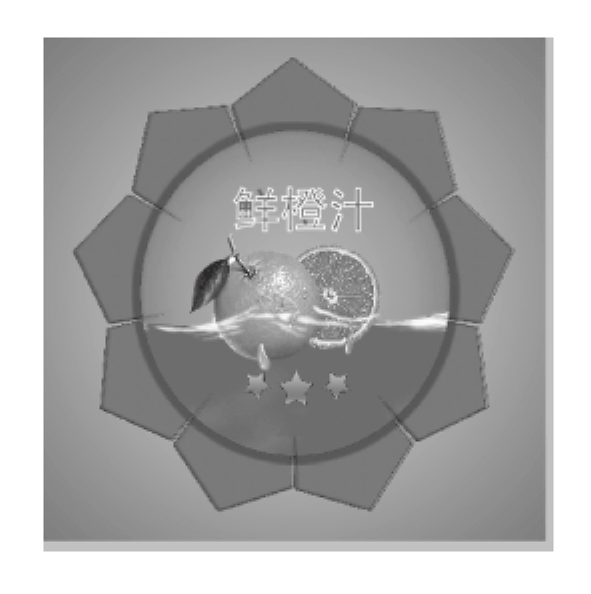


图7-39　设置文字



图7-40　文字变形

(6) 保存、预览。

第8章 基础动画的制作

本章学习目标：

✍了解动画的基础知识。

✍掌握逐帧动画的制作。

✍掌握各种过渡动画的制作。

8.1 动画的基础知识

动画，可以说是一种老少皆宜的艺术形式，它是通过播放一系列画面，给视觉造成连续变化的图画，它的基本原理与电影一样，都是利用人类的视觉暂留特性工作的。

8.1.1 帧

动画的形成原理是多张图片连续变换时，由于人眼具有视觉停留的特性，使人产生这些图片内容动起来的感觉，Flash CS6 也是根据这个原理产生动画效果的。在整个动画的制作过程中，帧是一个最基本的概念，贯穿于动画制作的全过程，动画就是通过改变连续帧的内容来实现的。

1. 帧的概念

电影是由一格一格的胶片按照先后顺序播放出来的，由于人眼有视觉停留现象，这一格一格的胶片按照一定速度播放出来看起来就"动"了，动画制作利用的也是这一原理，而 Flash 中的"帧"就相当于一格一格的胶片。

2. 帧的分类

在 Flash CS6 中动画制作是通过改变连续帧的内容来实现的。首先要了解帧的分类，帧一共分为 3 类，即关键帧、过渡帧和空白关键帧。

1）关键帧

关键帧在制作动画的过程中是非常关键的，只有定义了关键帧和动画的属性，才能让 Flash 自动完成动画过程。关键帧就是定义动画的起始点和终结点，也就是说，动画的起始点和终结点都是关键帧。

2）过渡帧

在定义了 Flash 的起始关键帧和终结关键帧之后，在起始和终结关键帧之内的帧被称为过渡帧。过渡帧是具体动画实现的详细过程图，它具体体现了动画变化过程。当单击过渡帧时，在工作区将预览这一帧的动画情况。

3）空白关键帧

在一个关键帧里面什么对象也没有添加，这种关键帧称为空白关键帧。单击鼠标左键，选定一个影格，然后右击，在弹出的快捷菜单中选择【插入空白关键帧】命令，这样的关键帧就是空白关键帧。

3. 帧操作

1）插入帧

- 选择菜单栏中的【插入】|【时间轴】|【帧】命令或按 F5 键，可以在时间轴上插入一个普通帧。
- 选择菜单栏中的【插入】|【时间轴】|【关键帧】命令或按 F6 键，可以在时间轴上插入一个关键帧。
- 选择菜单栏中的【插入】|【时间轴】|【空白关键帧】命令，可以在时间轴上插入一个空白关键帧。

2）选择帧

- 选择菜单栏中的【编辑】|【时间轴】|【选择所有帧】命令，可以选中时间轴上的所有帧。
- 在时间轴上单击要选择的帧，帧变为蓝色。
- 在按住 Ctrl 键的同时用鼠标单击要选择的帧，可以选择多个不连续的帧。
- 在按住 Shift 键的同时用鼠标单击要选择的两个帧，可以选择这两个帧之间的所有帧。

3）删除帧

选中时间轴上要删除的帧，选择菜单栏中的【编辑】|【时间轴】|【删除帧】命令或按 Shift+F5 键，都可以将选定的帧删除。在一个帧被删除后，后面的帧将自动向前移动。

4）移动帧

在选中一个帧或多个帧后，被选中的帧反亮显示，按住鼠标左键移动所选帧到目标位置。

8.1.2 时间轴与图层

1. 时间轴

时间轴是 Flash 编辑动画的主要工具，是用于组织和控制动画中的帧和层在一定时间内播放的坐标轴。

【时间轴】面板由【图层】面板和时间轴组成，如图 8-1 所示。

【图层】面板的中间部分是图层显示区，文档中的所有图层都会在此处显示，上面的图层覆盖下面的图层，图层类型等与图层相关的内容都会在此处显示。

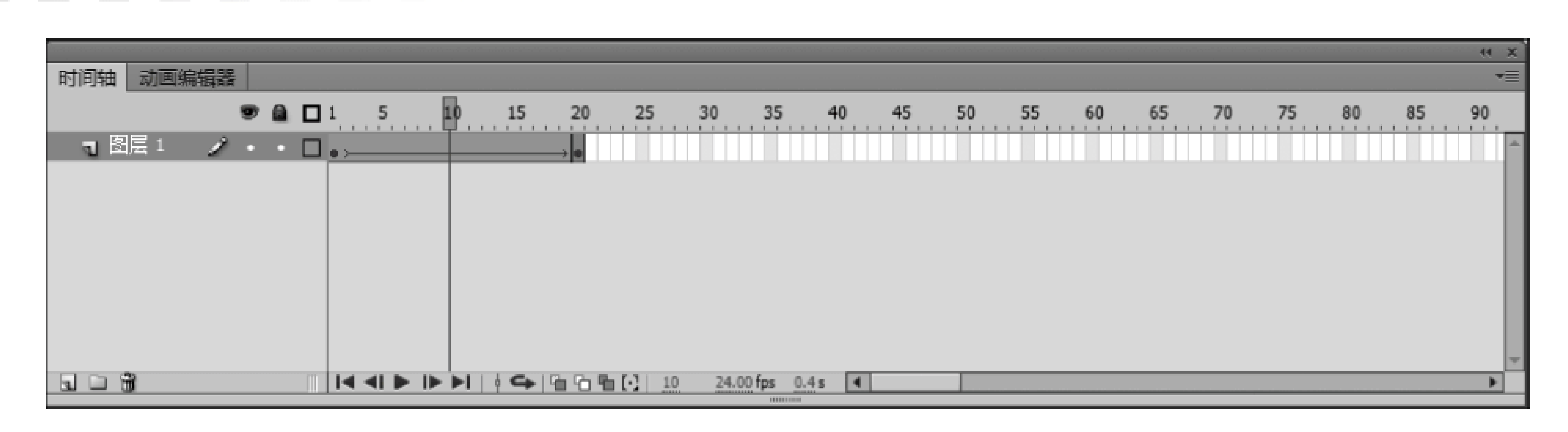


图 8-1 【时间轴】面板

- 【眼睛】图标：单击此图标，可以显示/隐藏图层中的内容。
- 【锁状】图标：单击此图标，可以锁定或解锁图层。
- 【线框】图标：单击此图标，可以将图层中的内容以线框的方式显示。
- 【新建图层】按钮：用于创建新图层。
- 【新建文件夹】按钮：用于创建图层文件夹。
- 【删除】按钮；用于删除无用的图层。
- 右侧时间轴中间部分对应图层显示各图层应用的帧，即各图层的动画状态。红色为播放头，它所在的位置即当前播放位置，也就是当前显示的那一帧。
- 【帧居中】：把当前的帧移动到时间轴窗口的中间，以方便操作。
- 【循环】：在规定帧内循环播放。
- 【绘图纸外观】：同时查看当前帧与前后若干帧里的内容，以方便前后多帧对照编辑。
- 【绘图纸外观轮廓】：在标记范围内的帧上的对象将以轮廓线的形式同时显示在舞台上。
- 【编辑多帧】：单击此按钮，绘图纸标记范围内的帧上的对象将同时显示在舞台中，可以同时编辑所有的对象。
- 【修改标记】：单击此按钮会弹出下拉菜单，其中有 5 个命令，如图 8-2 所示。
  - 【始终显示标记】命令：在时间轴标尺上总是显示出绘图纸标记。
  - 【锁定标记】命令；将锁定绘图纸标记的显示范围，移动播放头不会改变显示范围。
  - 【标记范围 2】命令：绘图纸标记显示范围从当前帧的前两帧开始，到当前帧的后两帧结束。
  - 【标记范围 5】命令：绘图纸标记显示范围从当前帧的前 5 帧开始，到当前帧的后 5 帧结束。
  - 【标记整个范围】：绘图纸标记显示范围为时间轴中的所有帧。

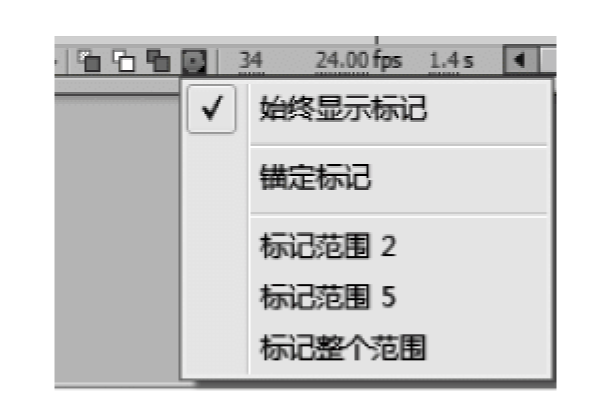


图 8-2 修改标记下拉菜单

- 【当前帧】：动画正在播放的那一帧。
- 【帧速率】：动画播放的速率，即每秒钟播放的帧数(fps)，可以打开文档【属性】面板进行设置，默认值为 12fps。
- 【动画时间】：当前动画所用的时间。

2. 图层

图层在 Flash 中是一个相当重要的动画制作手段,尤其是在创建复杂的动画时具有很大的作用。利用 Flash CS6 的图层技术可以创建许多动画的特殊效果,并且对一个图层上的对象进行编辑、粘贴和重新定位的操作不会影响到其他图层。

那么什么才是图层呢?一个图层犹如一张透明的纸,在上面可以绘制任何事物或书写任何文字,所有的图层叠在一起,就组成了一幅完整的画。层有两大特点:除了画有图形或文字的地方,其他部分都是透明的,也就是说,下层的内容可以通过透明的这部分显示出来;图层又是相对独立的,修改其中的一层不会影响到其他层。使用图层最大的好处就是可以将不同类型的内容放在不同的图层中,便于管理。

图层位于【时间轴】面板的左边,如图 8-1 所示。图层是一层一层向上叠的,在默认情况下新建的图层位于旧的图层之上,不过可以通过使用鼠标按住上下拖动来修改这种排列,在图层名称上双击可以直接修改名称。

图层可以分为普通图层、遮罩图层和引导图层,其中引导图层又分为普通引导图层和运动引导图层两种。

8.1.3　元件的创建与编辑

元件是指在 Flash 创作环境中或使用 Button(AS 2.0)、SimpleButton(AS 3.0)和 MovieClip 类创建过的图形、按钮或影片剪辑,然后可以在整个文档或其他文档中重复使用,元件可以包含从其他应用程序中导入的插图。用户创建的任何元件都会自动成为当前文档的库的一部分。

在 Flash 中所产生的一切运动(形状变形除外)都要以元件的形式来实现,所以在设计运动动画之前要将编辑好的对象转化为元件。使用元件来设置动画除了方便管理和调试外,最重要的是如果一个对象要重复使用,只需多次从库中调用,且不会增加文件的体积。

元件就像是给图形套了一件衣服,然后衣服有了分身术的魔法,让用户可以无限制地让同一个图形多次出现在舞台上,而且可以随便改变大小,图形的清晰度是不变的。

在文档中使用元件可以显著减小文件的大小,保存一个元件的几个实例比保存该元件内容的多个副本占用的存储空间小。例如,通过将背景图像这样的静态图形转换为元件,然后重新使用它们,可以减小文件的大小。使用元件还可以加快 SWF 文件的回放速度,因为元件只需下载到 Flash Player 中一次。

在创作或运行时,可以将元件作为共享库资源在文档之间共享。对于运行时共享资源,可以把源文档中的资源链接到任意数量的目标文档中,而无须将这些资源导入到目标文档中。对于创作时共享的资源,可以用本地网络上可用的其他任何元件更新或替换一个元件。

1. 元件与实例

元件是 Flash 中最重要的也是最基本的元素,它在 Flash 中对文件的大小和交互能力起着重要的作用。元件是位于当前动画库中可以反复使用的图形、按钮、动画或声音资源。

制作的元件或从外部导入的文件都会保存在库中。在使用元件的时候可以将元件从库中拖至工作区或其他元件中，这些被拖出来的元件称为实例，如图 8-3 所示。

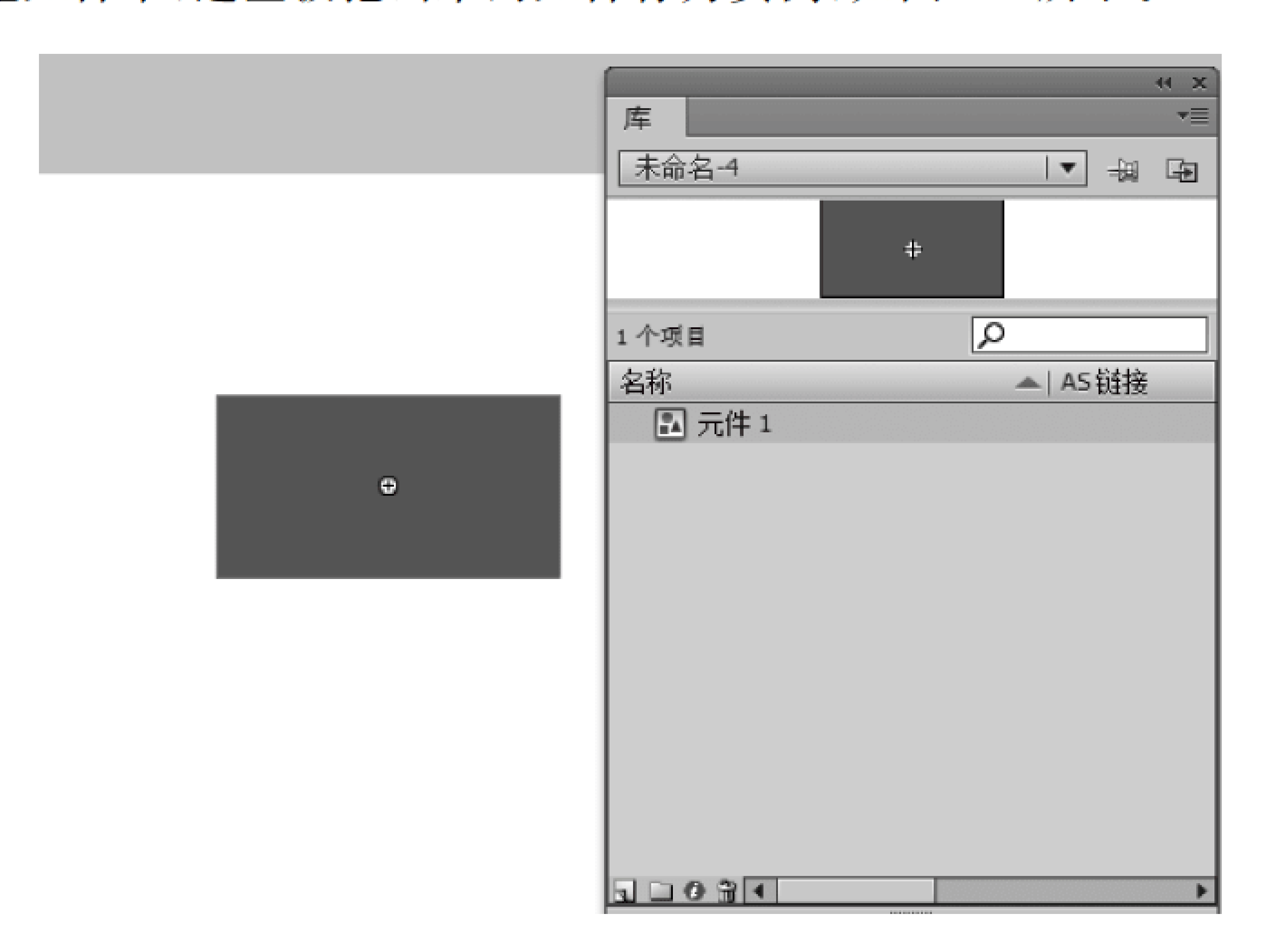


图 8-3　元件 1 与其实例

元件的应用会使动画制作十分方便、轻松，在制作一个动画的过程中，如果对使用的图像元素重新进行编辑，那么还要对使用了该图像元素的复制品进行编辑。如果使用了元件，就不会出现这样的重复操作了，只要将在动画中重复出现的元素制作成元件，当使用元件时，将元件从库中拖到工作区就可以了。当元件被重新编辑后，所有的实例都会随之改变。

除此之外，合理地使用元件还可以减小影片文件的大小。在制作影片的过程中，如果每次都使用独立的图形对象，Flash 将存储所有图形对象的信息，这样会使动画的大小非常大，但是如果将图形对象制作成元件，无论工作区中有多少个同一元件的实例，在 Flash Player 中都只载入一次元件内容，所以使用了元件的影片文件会很小。例如，在工作区中应用一个元件，输出后的文件大小为 5KB，在工作区中应用 3 个相同的元件，输出的文件大小仍为 5KB，但是如果在工作区中拖入 3 个图形，则输出文件的大小为 15KB。

同一个元件在不同的实例中可以有一些不同的属性，如大小、颜色、旋转的角度、透明度等，改变实例中的属性并不影响库中的元件；即使用户对实例进行修改，Flash 仅需要存储一些与元件有差别的信息，因此可以使影片文件的输出大小最小。

2. 元件的分类

根据功能和使用环境的不同，元件共有 3 种形式，即图形、按钮和影片剪辑。

图形元件表示的是用于创建可反复使用的图形，它可以是静止图片，也可以是由多个帧组成的动画。它的特点是拥有相对独立的编辑区域，如果将其调用到场景中，会受到场景帧约束。

按钮元件用于创建动画的控制按钮，以响应鼠标的按下、单击等事件。按钮元件包括“弹起”、“指针经过”、“按下”和“点击”4 种状态，在按钮元件的不同状态上创建不同的内容，

每种状态都可以通过图形、声音来定义。在创作按钮元件时，还可以设计出各种互动效果，使按钮对鼠标操作进行相应的响应。

影片剪辑元件表示的是影片中的小片段，它可以是静态的图形，也可以是一段动画，可以在影片中增加ActionScript、动画、声音和其他影片。每个影片都有自己独立的时间轴，可以独立播放。当播放主动画时，影片元件也在循环播放，它不会受到场景中帧的约束。

3. 元件的创建

创建元件的方法有两种，一种是直接创建，另一种是转换。

1）直接创建

选择菜单栏中的【插入】|【新建元件】命令（或按Ctrl+F8键），打开【创建新元件】对话框，在该对话框的【名称】文本框中输入创建元件的名称，在【类型】选项区中可以选定元件的类型，单击【确定】按钮后，进入图形元件的编辑状态，绘制相应的图形，单击工作区中的标题栏“场景1”切换到主场景中，【库】面板中就出现了创建的图形元件。

2）将对象转换为元件

使用工具箱中的选择工具选中对象，然后选择菜单栏中的【修改】|【转换为元件】命令（或者按F8键），在打开的【转换为元件】对话框中输入名称，选择元件的类型，单击【确定】按钮，此时【库】面板中就会出现刚转换的元件。

4. 元件的编辑

元件的编辑可以在多种模式下进行，一种是一般的元件编辑模式，另一种是在当前环境下进行编辑，还有一种是在新窗口中进行编辑。

1）在元件编辑模式下进行编辑

选中要编辑的元件，然后右击，在弹出的快捷菜单中选择【编辑】命令，进入元件编辑模式。

2）在当前编辑模式下进行

右击元件实例，在弹出的快捷菜单中选择【在当前位置上编辑】命令，在当前窗口中其他对象变灰，被编辑元件的名称出现在舞台顶端的信息栏中。

3）在新窗口中编辑

在场景中右击某实例，在弹出的快捷菜单中选择【新建窗口编辑】命令，被编辑的元件名称将出现在舞台顶端的信息栏中。

5. 3种元件的相同点与区别

1）相同点

3种元件的相同点是都可以重复使用，且当需要对重复使用的元素进行修改时只需编辑元件，而不必对所有该元件的实例一一进行修改，Flash会根据修改的内容对所有元件的实例进行更新。

2）区别及应用中需注意的问题

（1）影片剪辑元件、按钮元件和图形元件最主要的差别在于影片剪辑元件和按钮元件

的实例都可以加入动作语句,图形元件的实例则不能;影片剪辑元件的关键帧可以加入动作语句,按钮元件和图形元件则不能。

(2) 影片剪辑元件和按钮元件中都可以加入声音,图形元件则不能。

(3) 影片剪辑元件的播放不受场景时间线长度的制约,它有元件自身独立的时间线;按钮元件独特的4帧时间线并不自动播放,只是响应鼠标事件;图形元件的播放完全受制于场景时间线。

(4) 影片剪辑元件在场景中按回车键测试时看不到实际播放效果,只能在各自的编辑环境中看效果,而图形元件在场景中可即时观看,可以实现所见即所得的效果。

(5) 3种元件在舞台上的实例都可以在【属性】面板中相互改变其行为,也可以相互交换实例。

(6) 在影片剪辑元件中可以嵌套另一个影片剪辑元件,在图形元件中也可以嵌套另一个图形元件,但在按钮元件中不能嵌套另一个按钮元件,3种元件可以相互嵌套。

8.1.4 场景概述

在设计Flash动画时,如果设计的动画比较复杂,动画时间较长,会导致图层的操作非常烦琐,此时可以将动画分成几段,分别放置在不同的场景中来制作。

1. 场景的概念

场景可以看成是一段相对独立的动画,当动画中有多个场景时,整个动画会按照顺序播放,当然,也可以用脚本程序来控制场景的播放顺序。

在Flash中,所有的动画制作都是在"场景"中实现的。当新建Flash文档时,Flash自动创建一个名称为"场景1"的场景。如果设计的是一个简单的小动画,只需要使用一个场景就可以了,如果设计的是一个比较复杂的动画,一个场景是不够的,可能需要多个场景。例如,网上的大部分Flash作品在开始的时候都有一个Loading场景来显示下载的速度。

2. 场景的创建

选择菜单栏中的【插入】|【场景】命令,可以创建一个新的场景。

3. 编辑场景

选择菜单栏中的【窗口】|【其他面板】|【场景】命令,可以打开【场景】面板,如图8-4所示。

在【场景】面板中可以对场景进行复制、添加、重命名、删除、更改播放顺序等操作。



图8-4 【场景】面板

8.1.5 动画的类型

以前制作动画的过程非常复杂,需要将每一帧上的画面全部绘制出来,有了Flash之后

制作动画变得简单了，我们只需要将动画画面中的关键部分绘制出来，中间的动画过程可以交给 Flash 完成，从而大大减少了制作动画的工作量。因此，在 Flash 动画的制作过程中，根据制作方法和生成原理的不同，可以将 Flash 动画分成逐帧动画和补间动画两种，而补间动画又可以分为动画补间动画和形状补间动画两种。

(1) 逐帧动画：逐帧动画是一种传统的动画形式，制作起来比较烦琐。此类动画完全由关键帧组成，每个关键帧都可以进行单独编辑，它需要用户对物体运动的基本规律有深的了解，并且具有一定的绘图基础。利用逐帧动画可以获得任意的动画效果，但是由于每个关键帧都要进行编辑，因而逐帧动画的工作量很大，而且制作出来的作品文件容量也很大。

(2) 动画补间动画：动画补间动画可以实现对象的移动、缩放、旋转、变色等动画效果，构成动画补间动画的元素是元件。

(3) 形状补间动画：形状补间动画可以完成图形的移动、缩放、形状渐变、色彩渐变等，它的作用对象只能是图形。和一般的动画补间动画有所不同，在使用形状变形时不需要把绘制好的形状转换为元件，如果要使用元件实例创建形状补间动画，需要将该实例通过按 Ctrl＋B 键分离后才可以实现。

8.2　逐帧动画

逐帧动画是 Flash 动画中最基本的动画形式，用户只要掌握了插入关键帧的方法就可以制作逐帧动画，但制作一个好的逐帧动画需要掌握物体的运动规律和具有好的绘画基础。

8.2.1　直接导入生成逐帧动画

用户可以将绘制好的 JPG、PNG 等格式的静态图片连续导入到 Flash 中来建立一段逐帧动画，也可以通过导入 GIF 序列图像、.swf 动画文件等来建立一段逐帧动画。

8.2.2　创建逐帧动画

在 Flash 中，从第 1 帧开始就建立关键帧，然后利用鼠标或者手写板在 Flash 中绘制一帧帧的矢量图形来生成逐帧动画，也可以用文字来做帧中的对象，实现文字的跳跃、闪烁等特效。图 8-5 所示为逐帧动画的时间轴。

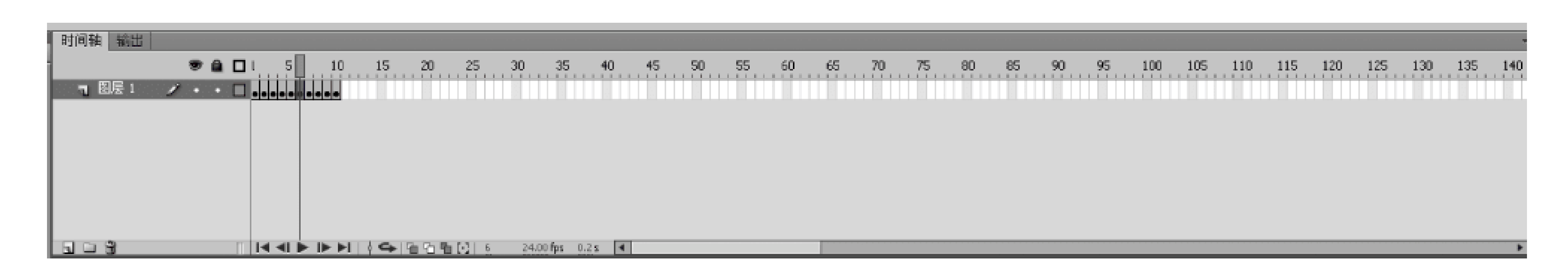


图 8-5　逐帧动画的时间轴

8.3 动画补间动画

补间动画是相对于逐帧动画而言的，它是事先定义好动画的起点和终点的内容，然后由Flash自动完成两点之间的动画过渡。与逐帧动画相比，补间动画不需要一帧一帧地绘制动画，避免了逐帧动画制作烦琐、容量过大等缺点。

动画补间动画是Flash中最常用的一种动画形式，使用动画补间动画可以制作出对象的移动、旋转、色彩变化、放大、缩小、透明度变化等动画，在动画补间动画中使用的对象只能是元件，包括影片剪辑元件、图形元件、按钮元件等，除了元件，其他元素都不能创建动画补间动画，矢量图、位图等都必须要转换成元件才可以用来制作动画补间动画。

在制作动画补间动画时，每一段动画中只能使用一个对象，如果在同一时间要实现多个不同对象的运动，可以使用多个层来实现多个对象的动画补间。

8.3.1 动画补间动画的制作

在Flash CS6中创建动画补间动画的方法如下：

在动画的两个关键帧之间任意选择一帧，然后右击，在弹出的快捷菜单中选择【创建传统补间】命令，就可以创建出动画补间动画。动画补间动画的过渡帧底纹为浅紫色，显示箭头，从开始帧指向结束帧，如图8-6所示。

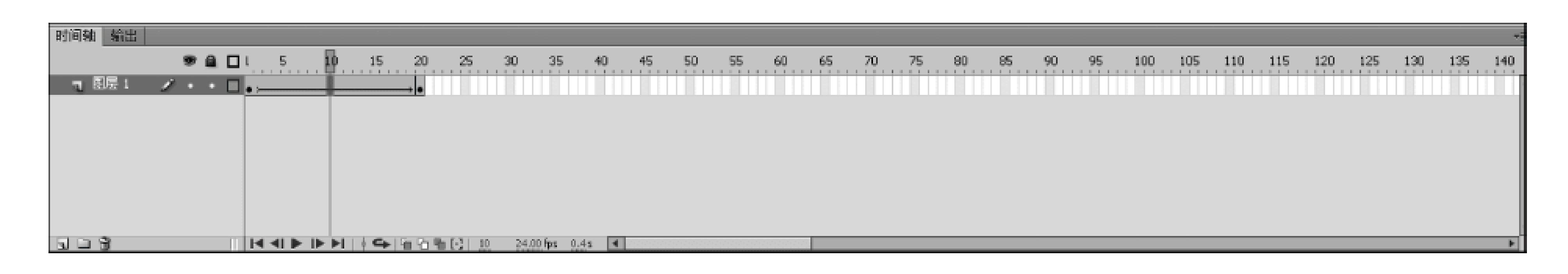


图8-6 动画补间动画的时间轴

8.3.2 动画补间动画的参数设置

在创建动画补间动画后，在【时间轴】面板上单击动画的起始帧，打开【属性】面板，在其中可以看到图8-7所示的选项。

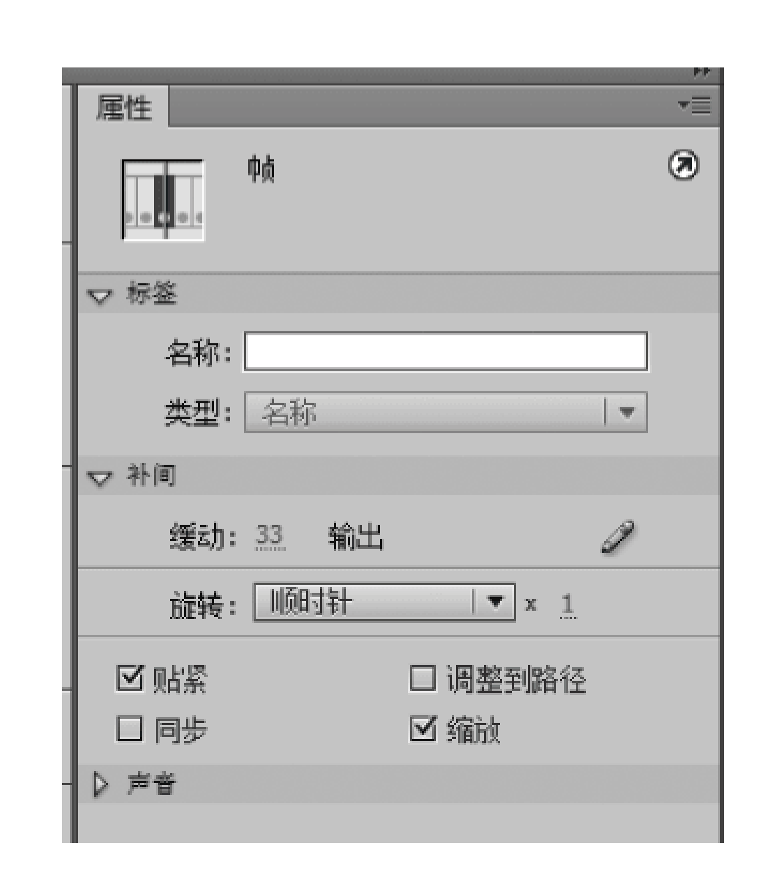


图8-7 动画补间动画的【属性】面板

(1)【缓动】选项：该选项用于设置对象的运动方式，即加速运动或减速运动。在“0”旁边有一个滑动条，单击后上下拉动滑动条或填入具体的数据，动画补间动画会随之发生相应的变化。

① 输入−1～−100之间的数值，动画运动的速度由慢到快，向着运动结束的方向加速补间。

② 输入1～100之间的数值，动画运动的速度由快到慢，向着运动结束的方向减速补间。

③ 在默认情况下,补间帧之间的变化速率是不变的,即"0"。

(2)【旋转】选项:该选项用于设置运动对象的旋转方式,在其下拉列表框中有4个选项。

① 无:禁止元件实例旋转。

② 自动:使元件实例根据起始帧与结束帧上的放置角度自动进行旋转。

③ 顺时针:使元件实例在运动时顺时针旋转相应的圈数。

④ 逆时针:使元件实例在运动时逆时针旋转相应的圈数。

(3)【调整到路径】复选框:勾选该复选框,将对象的基线调整到运动路径,此项功能是用于运动引导动画的。

(4)【同步】复选框:勾选该复选框,可以使图形元件实例内部动画的播放和舞台中的动画播放同步。

(5)【贴紧】复选框:勾选该复选框,可以根据注册点将动画对象吸附到运动路径上,此项功能也是用于运动引导动画的。

(6)【缩放】复选框:勾选该复选框,对象在动画过程中可以改变比例。

8.4 形状补间动画

形状补间动画是补间动画的另一种表现形式,它的作用是使一种物体的形状变化为另外一种形状,在形状变化的同时颜色、位置等也可以随之变化。形状补间动画与动画补间动画的不同在于形状补间动画是对两个不同的图形对象进行补间,而动画补间动画是针对同一个对象的不同变化来进行补间,因此在形状补间动画中作用的对象只能是图形,所以在使用形状变形时无须也不能把绘制好的形状转换为元件,如果要使用现有的元件或文字创建形状补间动画,要先将元件实例或者文字按Ctrl+B键分离后才能实现。

虽然形状补间动画也能实现对象的移动、放大、缩小及色彩、透明度等变化,但是在通常情况下不会使用形状补间动画来实现这些变化,而是通过动画补间动画来实现,这是因为使用形状补间动画创建动画,Flash在保存时会记录每一个关键帧上的图形,文件体积较大,而动画补间动画通常使用的是元件,Flash只要记录一次就可以了,所以在制作动画时如果能用动画补间实现最好不要用形状补间实现。

8.4.1 形状补间动画的制作

在Flash CS6中创建形状补间动画的方法如下:

在动画的两个关键帧之间任意选择一帧,然后右击,在弹出的快捷菜单中选择【创建补间形状】命令,就可以创建出形状补间动画。其过渡帧呈浅绿色底纹,显示箭头,从开始帧指向结束帧,图8-8所示为形状补间动画的时间轴。

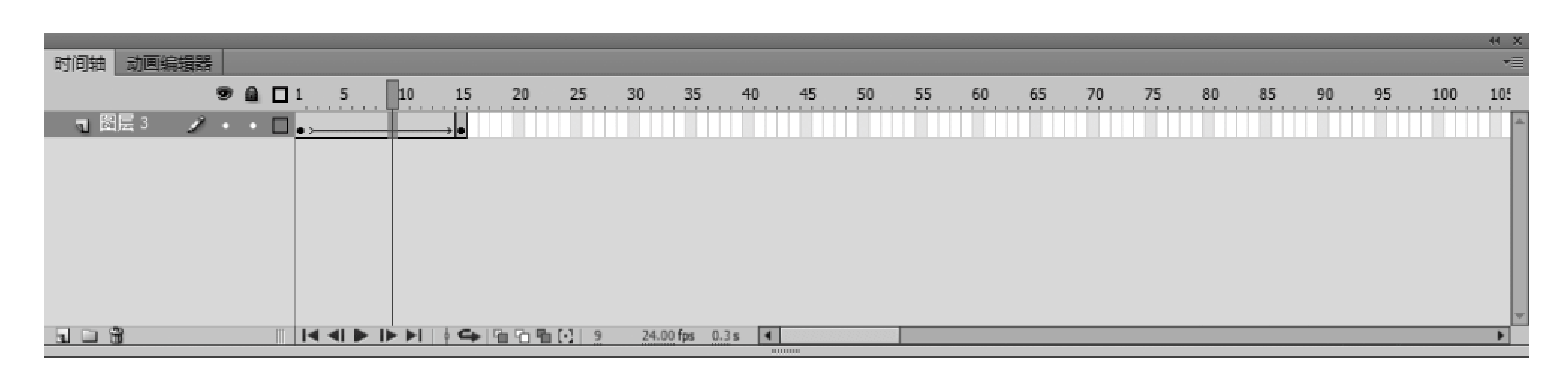


图 8-8　形状补间动画的时间轴

8.4.2　形状补间动画的参数设置

在创建形状补间动画后，在【时间轴】面板上单击动画的起始帧，打开【属性】面板，在其中可以看到如图 8-9 所示的选项。

(1)【缓动】选项：这个选项的作用和动画补间动画的一样，在此不再赘述。

(2)【混合】选项：作用是设置动画中间的形状变化模式。

- 分布式：使创建的动画的中间形状比较平滑和不规则。
- 角形：使创建的动画的中间形状保留明显的角和直线。

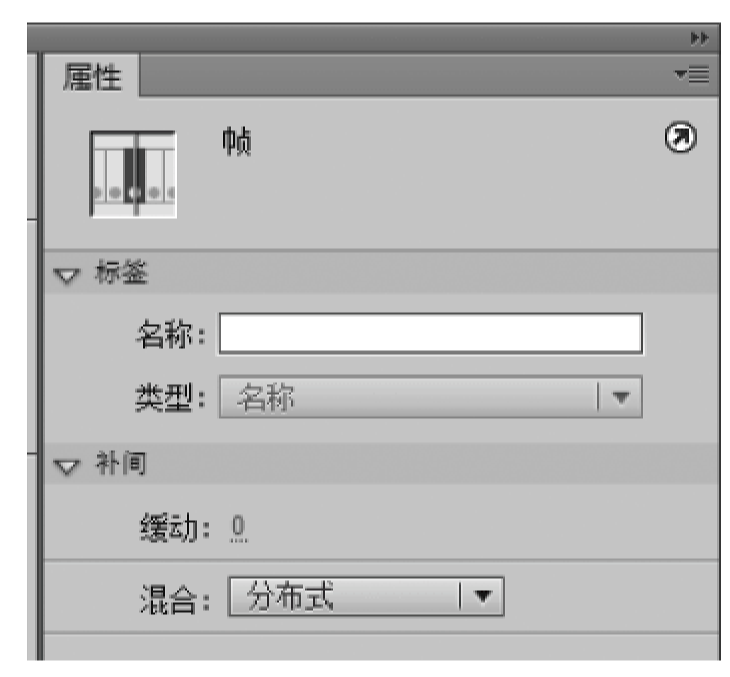


图 8-9　形状补间动画的【属性】面板

8.4.3　添加形状提示

虽然形状补间动画制作起来很简单，但是当前后图形差异比较大时变形效果较差。为了能使变形效果按照自己的意愿变化，可以使用【修改】|【形状】|【添加形状提示】功能，在起始帧和结束帧位置添加一些相应的参考点，使对象之间的变形不是随机的，而是有规律的，如图 8-10 所示。

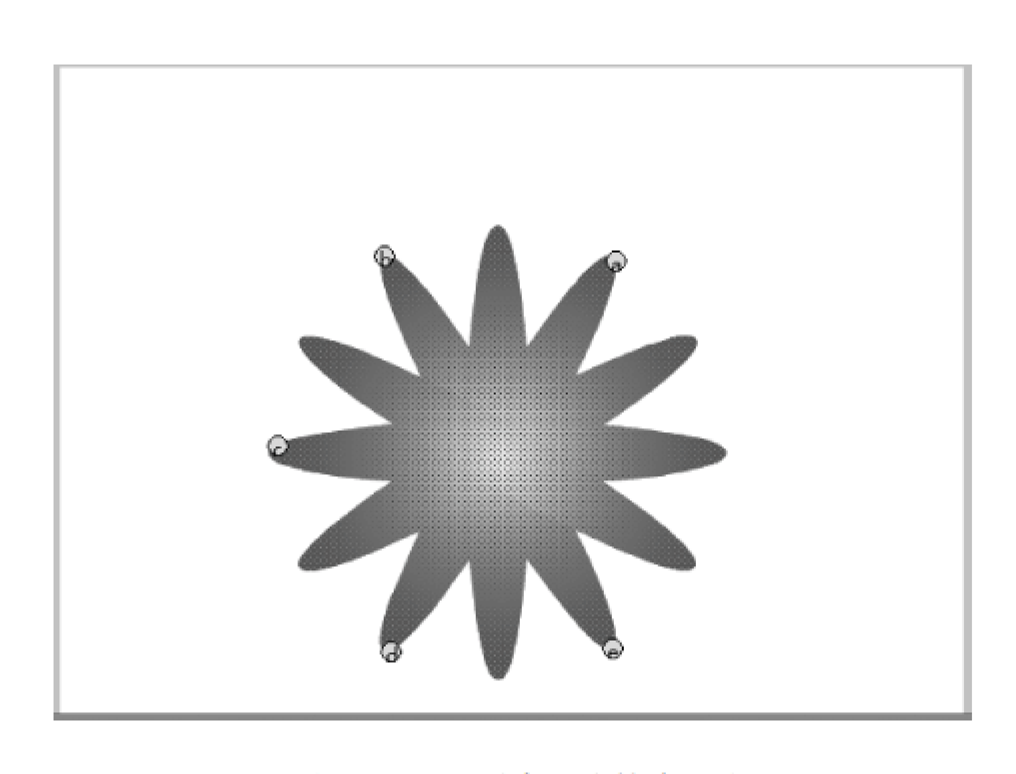


图 8-10　添加形状提示

8.5　综合应用

8.5.1　按钮的制作

制作按钮，灰色边框、白色的字，在指针经过帧变为绿色边框、红色的字，在按下帧为蓝色边框、黑色的字，操作步骤如下：

(1) 选择菜单栏中的【文件】|【新建】命令，创建文件，设置 200 像素×200 像素大小，其余参数默认，如图 8-11 所示。

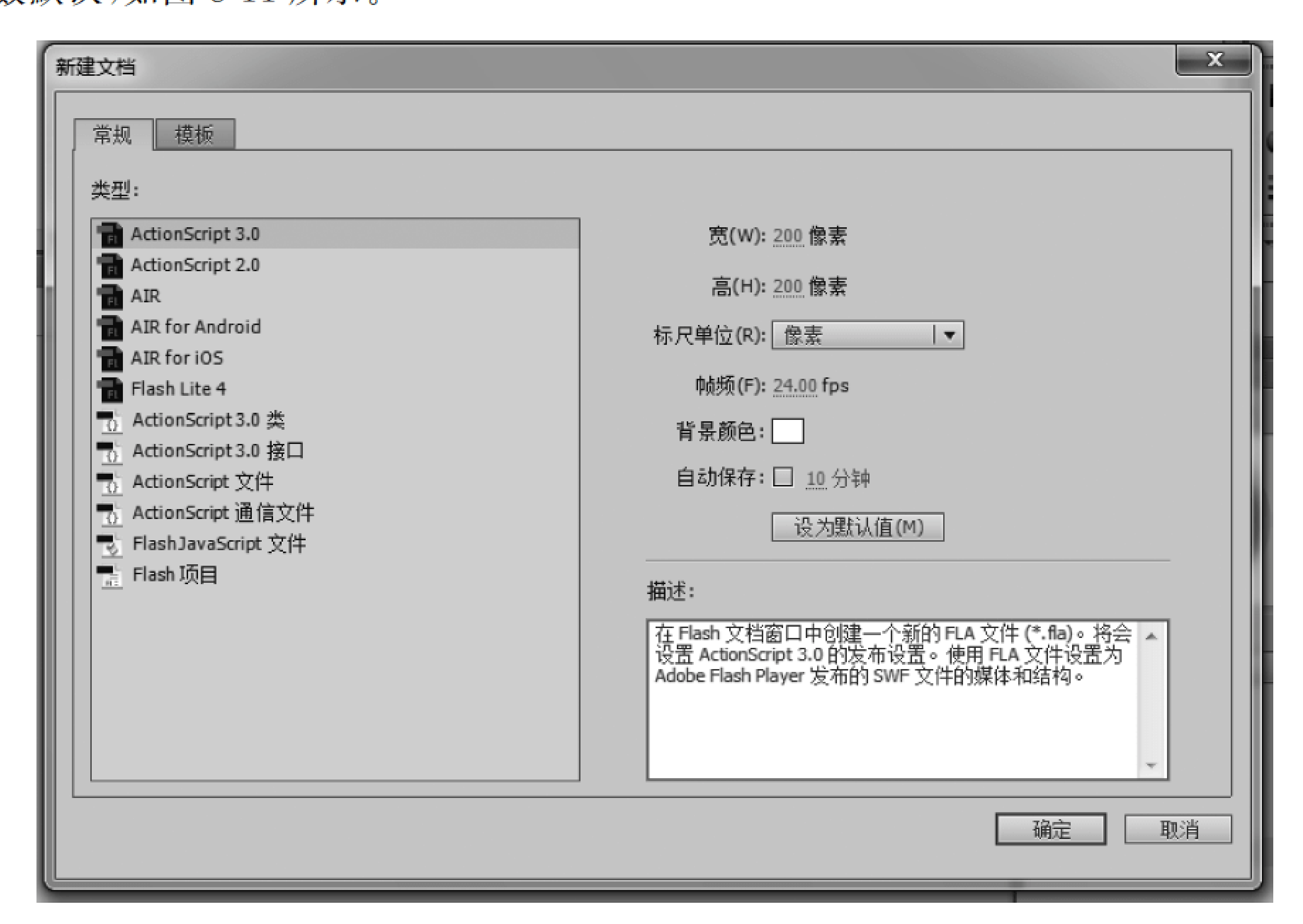


图 8-11　新建文件

(2) 选择菜单栏中的【插入】|【新建元件】命令，在【新建元件】对话框中选择类型为“图形元件”、设置元件名为“元件 1”，如图 8-12 所示。

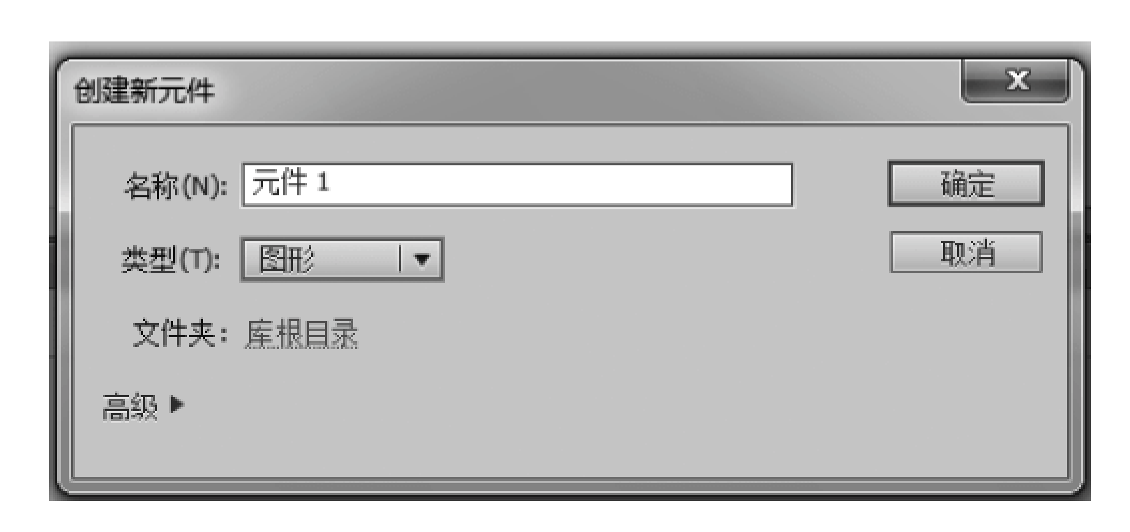


图 8-12　新建图形元件

(3) 选择椭圆工具，单击工具箱下方的【对象绘制】按钮，在元件编辑窗口中当光标呈“＋”字时从中心点按住 Shift＋Alt 键绘制正圆，然后选择【颜色】面板中的线性渐变，由灰到

白色对正圆进行填充。选择渐变编辑工具，将光标放在右上角点上旋转，调整渐变为从上到下，如图 8-13 所示。

(4) 重复步骤 3 的操作，绘制一个比灰圆小一点的圆，由浅绿到深绿填充，如图 8-14 所示。

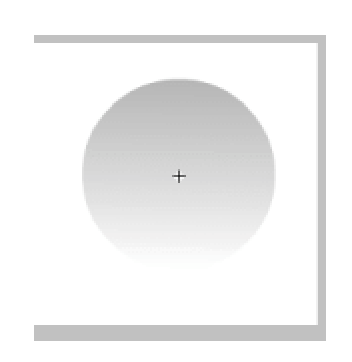




图 8-13　灰色圆的绘制和填充

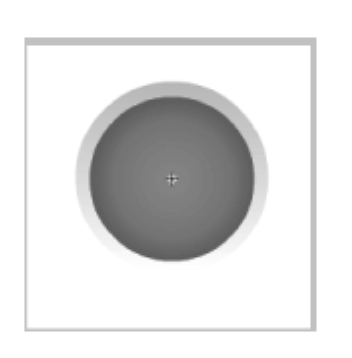


图 8-14　绿色圆的绘制和填充

(5) 再次选择椭圆工具，单击工具箱下方的【对象绘制】按钮，绘制一个小椭圆，然后选择【颜色】面板中的线性渐变，由灰到白对椭圆进行填充，将一侧的白色的透明度设为 0，并用渐变编辑工具调整渐变为从上到下，如图 8-15 所示。

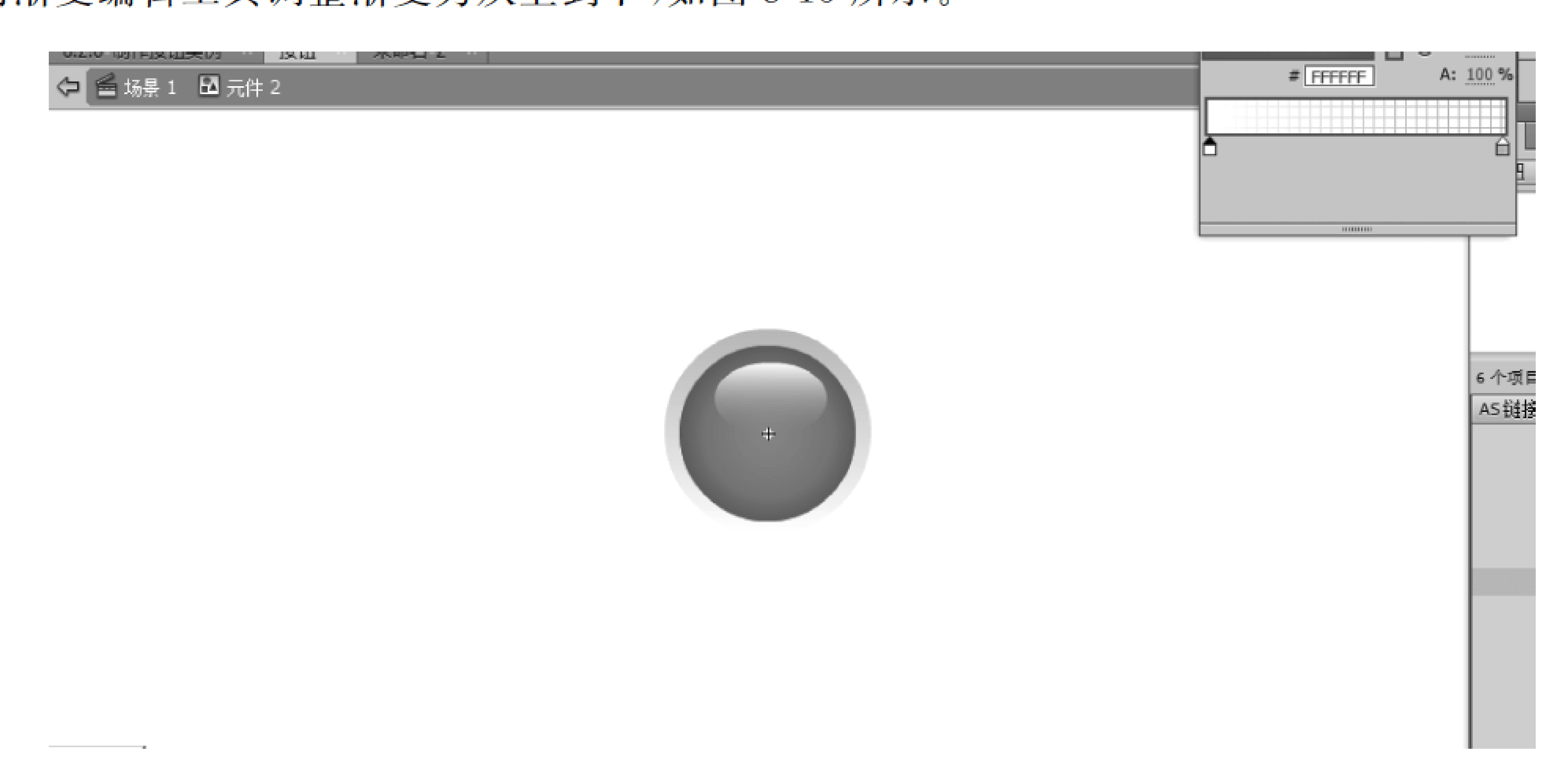


图 8-15　椭圆的绘制和填充

(6) 复制两次元件 1，得到“元件 1 副本”、“元件 1 副本 2”，将其中的颜色进行修改，将灰色的大椭圆分别改为浅蓝色和黄色，如图 8-16 所示。

(7) 选择菜单栏中的【插入】|【新建元件】命令，在【创建新元件】对话框中选择类型为“按钮”、设置元件名称为“元件 2”，如图 8-17 所示。

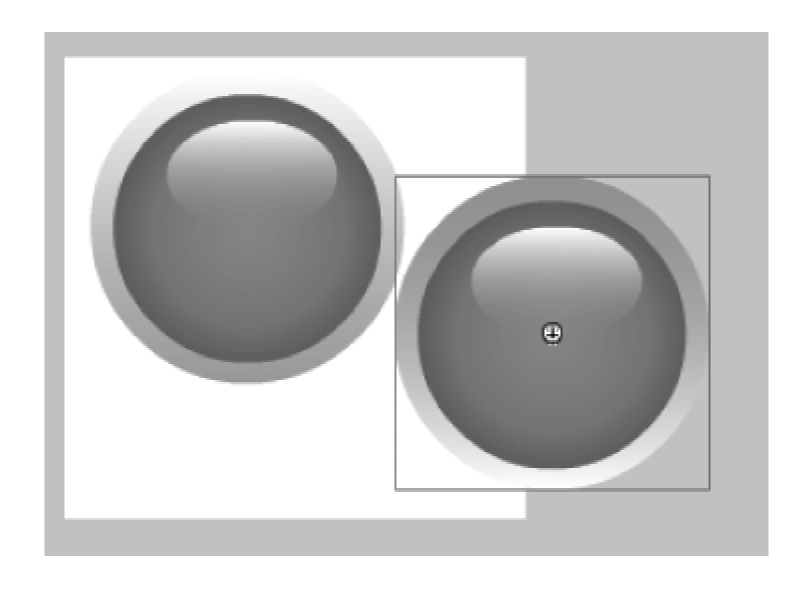


图 8-16　复制元件 1 并修改

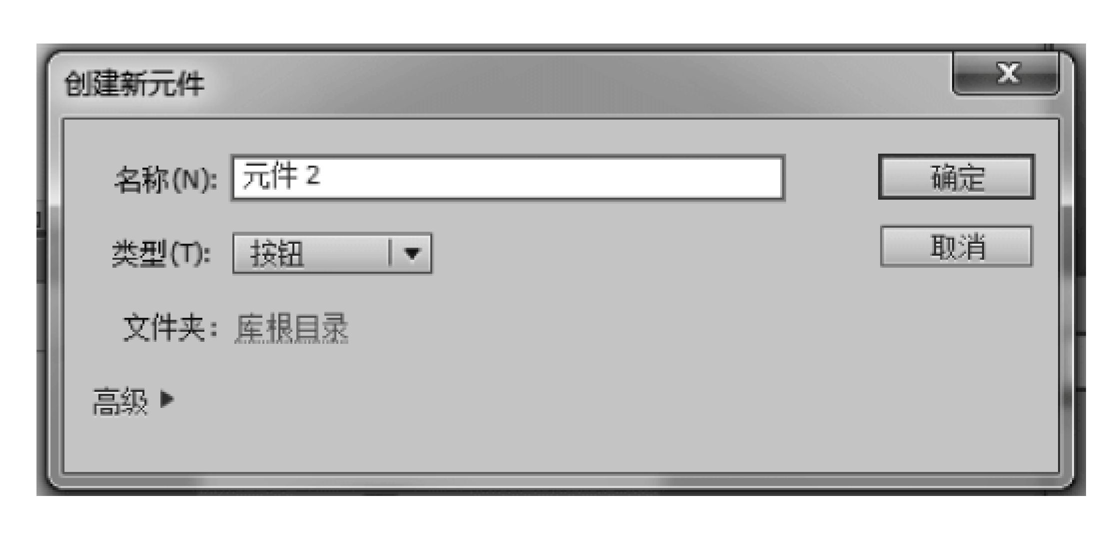


图 8-17　新建按钮元件

（8）在按钮元件的【时间轴】面板中对弹起、指针经过、按下 3 个帧插入关键帧，并将“元件 1”、“元件 1 副本”、“元件 1 副本 2”分别拖到这 3 帧，然后插入新图层，在新图层中分别为 3 个帧输入文字“弹起”、“滑过”、“按下”，并改变文字颜色为白色、红色、黑色，【时间轴】面板如图 8-18 所示。

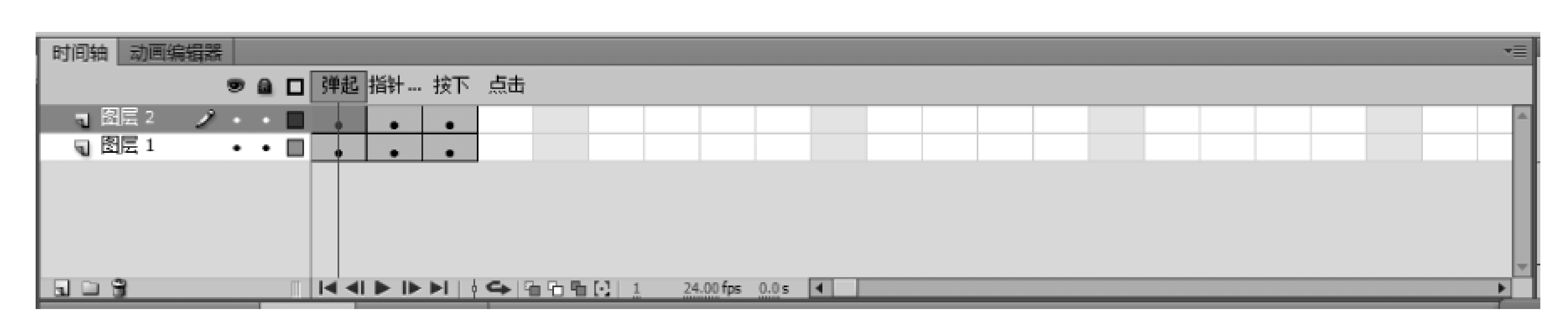


图 8-18　关键帧的【时间轴】面板

（9）将按钮拖入舞台，预览，效果如图 8-19 所示。



图 8-19　按钮实例效果图

8.5.2　运动小球的制作

制作一个运动的小球，让小球在天与地之间弹起、落下，碰到接触物时会变形为扁圆，其操作步骤如下：

（1）选择菜单栏中的【文件】|【新建】命令，建立一个 200 像素×600 像素大小的 Flash 文档，如图 8-20 所示。



图 8-20　新建文件

（2）选择菜单栏中的【文件】|【导入】|【导入到舞台】命令，将素材导入到文档中，移动到合适的位置，并按Ctrl+B键将图片打散。然后使用选择工具在素材中部拖动鼠标绘制矩形，并按Delete键将多余部分删除，如图8-21所示。

（3）单击【图层】面板下方的【新建图层】按钮创建“图层2”，然后选择椭圆工具，按住Shift键绘制一个正圆；选择线条工具，将笔触调为2绘制一段线段，用移动工具将其向下弯曲；复制线段，移动位置，形成图8-22所示的笑脸形状的小球。

（4）按F8键，在【转换为元件】对话框中将小球转换成图形元件，将其移动到天空的底部，然后在第15帧中插入关键帧，将小球移到绿地的位置，在第16帧中插入关键帧，选择任意变形工具，将小球压扁一点，在第17、18帧同理将小球压的更扁一些，在第19、20帧中插入关键帧，将小球调整回原始大小，如图8-23所示。

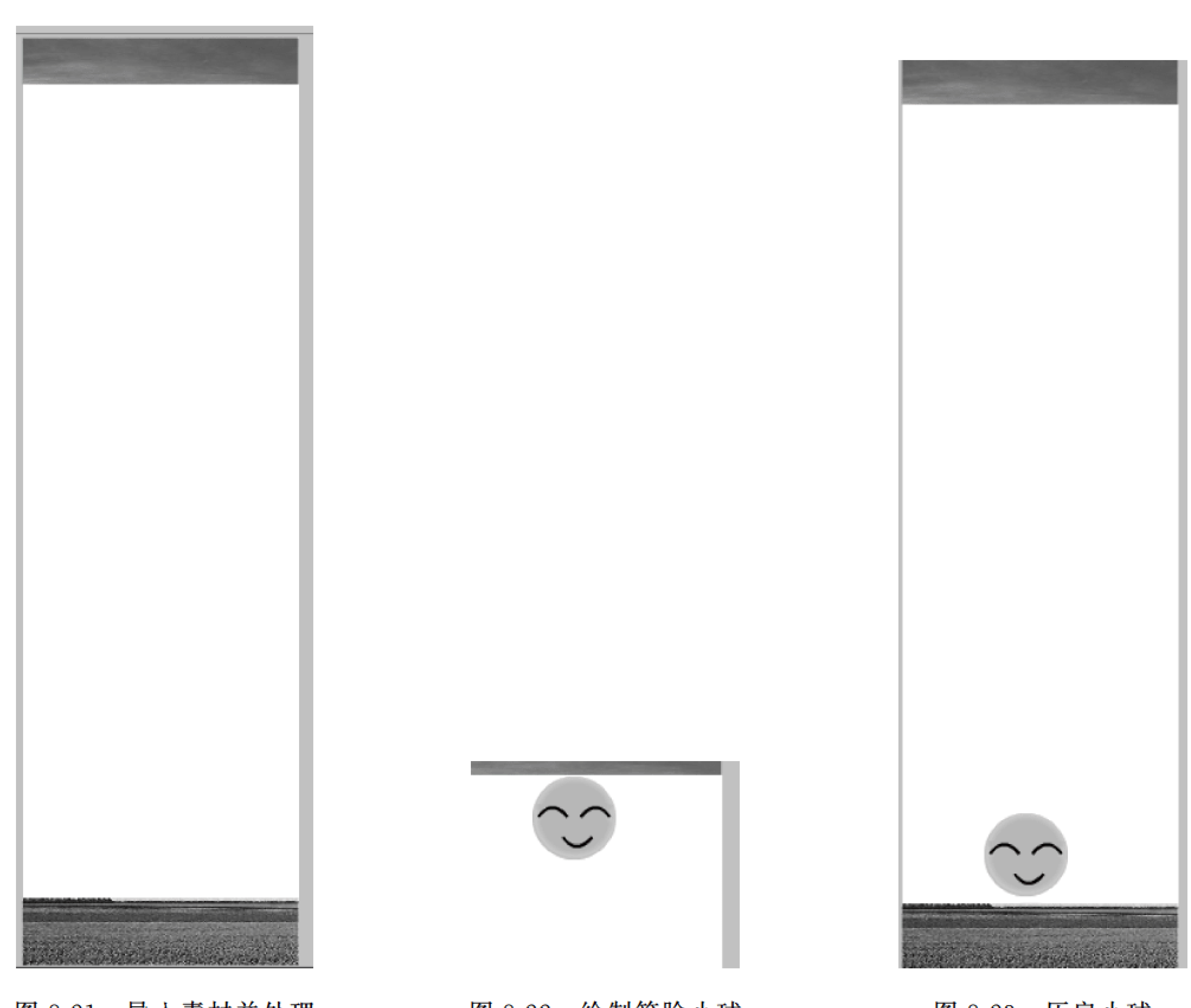


图8-21　导入素材并处理　　图8-22　绘制笑脸小球　　图8-23　压扁小球

（5）在第36帧中插入关键帧，将小球拖回天空位置，如图8-24所示；重复第4、5步，将小球压扁并再次恢复。将光标放在1～15帧之间的任意帧，然后右击，在快捷菜单中选择【创建传统补间】命令，时间轴上出现淡紫色底纹；同理，在20～36帧之间也创建传统补间。

（6）保存，预览，小球在天空与地面之间跳动，如图8-25所示。

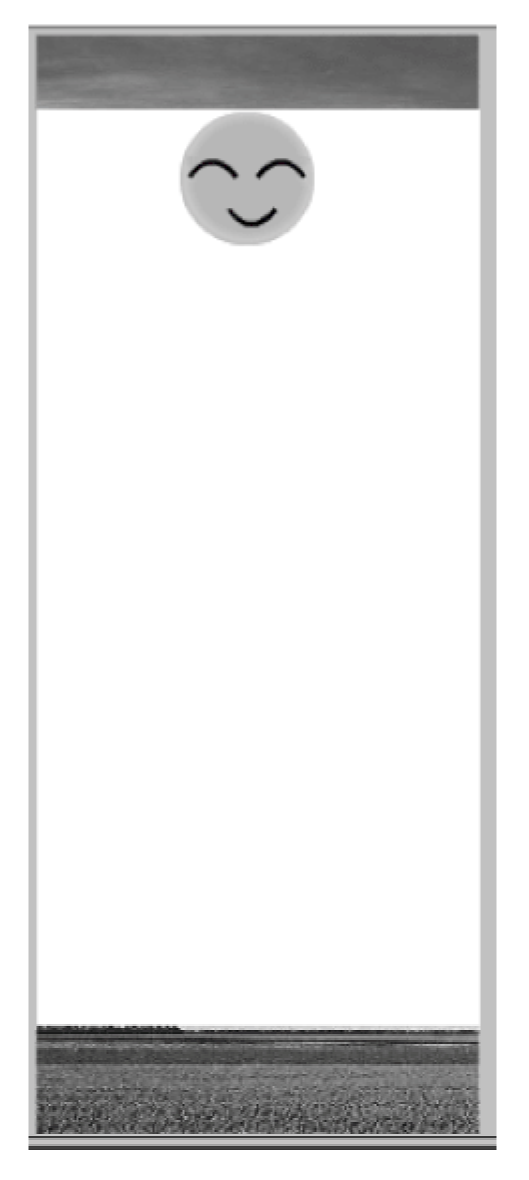


图 8-24　天空位置的小球



图 8-25　跳动的小球

8.5.3　欢迎光临 Banner 的制作

制作一个欢迎光临的 Banner，星星逐个进入舞台，变形为文字，其操作步骤如下：

(1) 选择菜单栏中的【文件】|【新建】命令，建立一个 480 像素×80 像素大小的 Flash 文档。

(2) 选择菜单栏中的【文件】|【导入】|【导入到舞台】命令，将“花”图片素材导入到舞台，并调整大小和位置，然后第 90 帧中插入帧，如图 8-26 所示。



图 8-26　导入图片“花”

(3) 选择菜单栏中的【文件】|【导入】|【导入到舞台】命令，将“星星”图片素材导入到舞台，并按 Ctrl+B 键打散。然后按 F8 键，将其转换为图形“元件 1”，如图 8-27 所示。

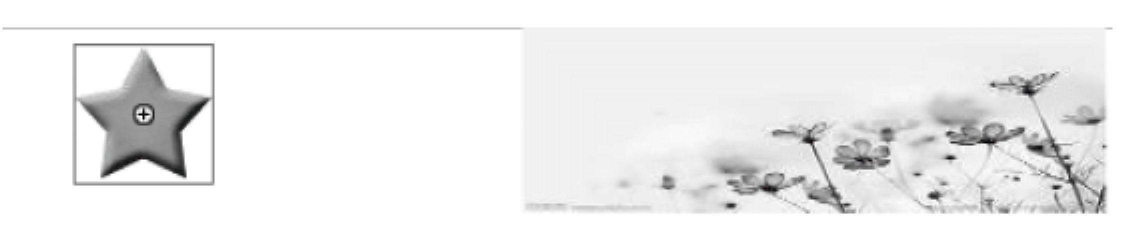


图 8-27　导入图片“星星”并转换为图形元件

(4) 单击【图层】面板下方的【新建图层】按钮，新建“图层 2”，然后单击第 1 帧，将“元件 1”拖到舞台外；在第 10 帧插入关键帧，将“元件 1”拖到舞台内的适当位置；在 1～10 帧之间的任意位置右击，在弹出的快捷菜单中选择【创建传统补间】命令；在第 43 帧插入关键帧，按 Ctrl+B 键将“元件 1”打散；在第 53 帧插入关键帧，将“元件 1”删除。选择工具箱中的文本工具，输入文字“欢”，在【属性】面板中设置大小为 60 点、颜色为红色、华文隶书，移到

和星星位置重合，并按 Ctrl+B 键将文字打散，同样，创建补间形状动画，如图 8-28 所示。



图 8-28 图层 2 的编辑

(5) 新建图层 3、4、5，在不同的帧位置重复第 4 步的操作，建立第 2、3、4 个星星的动画，【时间轴】面板如图 8-29 所示。

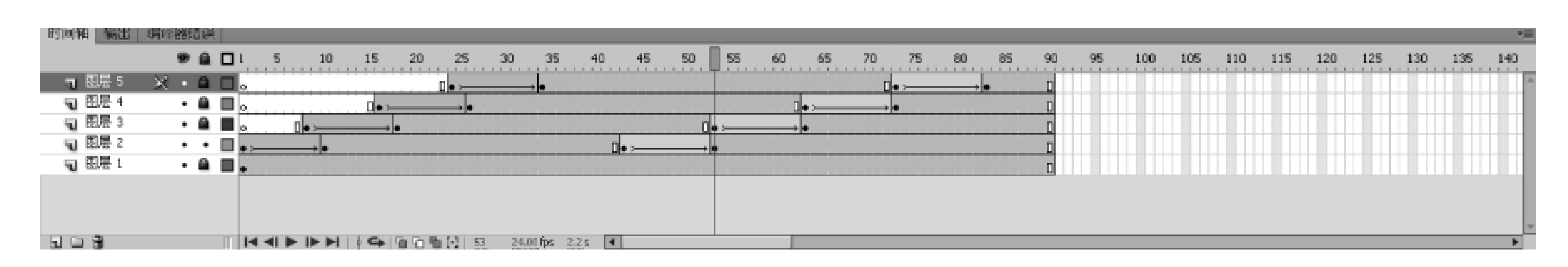


图 8-29 各图层的【时间轴】面板

(6) 保存，预览，最终效果如图 8-30 所示。



图 8-30 案例的最终效果

8.5.4 动画标识的制作

制作一个动画标识，3 个环一点一点地删除，箭头 3D 旋转。其操作步骤如下：

(1) 选择菜单栏中的【文件】|【新建】命令，建立一个 550 像素×400 像素大小的 Flash 文档，如图 8-31 所示。

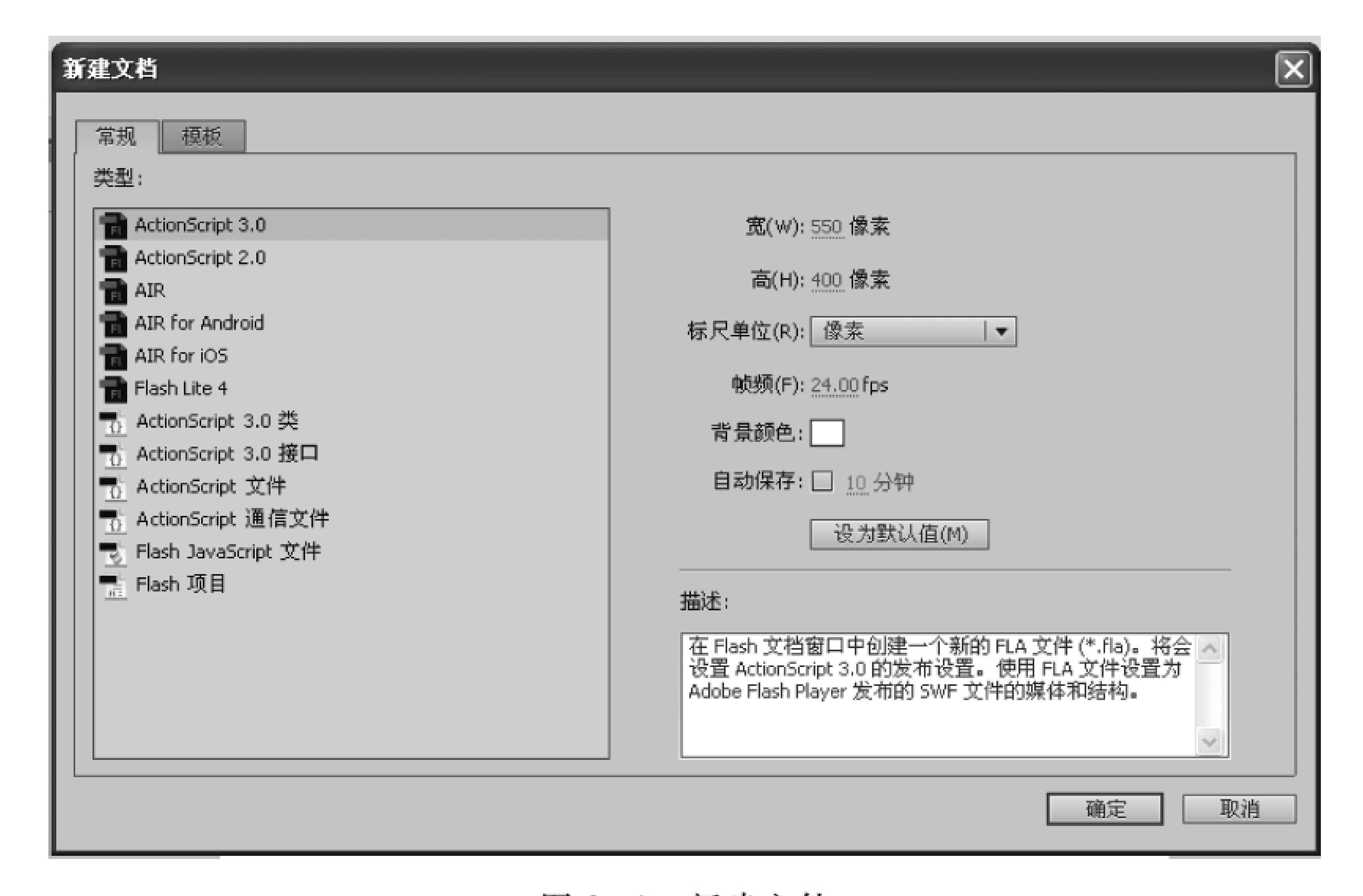


图 8-31 新建文件

（2）选择工具箱中的矩形工具，在舞台上拖动绘制一个550像素×400像素的矩形，填充颜色为#89E0D2，如图8-32所示。

（3）新建“图层2”，选择工具箱中的椭圆工具，按住Shift键在舞台上绘制一个正圆，并设置笔触大小为15、颜色为蓝色，如图8-33所示。



图8-32　绘制矩形

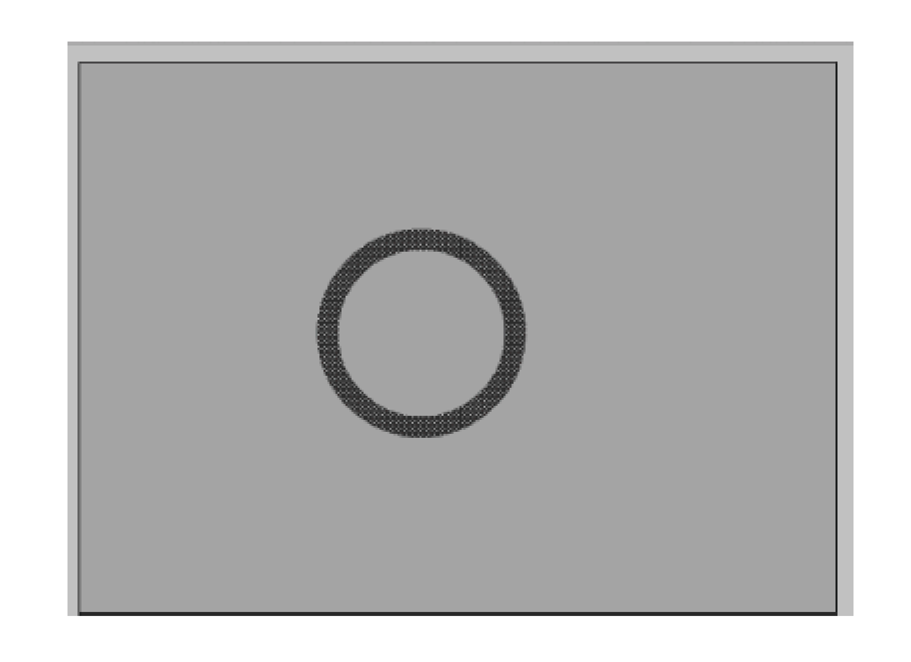


图8-33　绘制正圆

（4）新建“图层3”，按Ctrl+C键复制蓝色的圆，然后选择菜单栏中的【编辑】|【粘贴到当前位置】命令，将圆复制到“图层3”上与“图层2”相同的位置；选择工具箱中的任意变形工具，按住Shift+Alt键，在右上角控制点处拖动，成比例从中心点放大，并将笔触颜色改为红色，如图8-34所示。

（5）新建“图层4”，按Ctrl+C键复制蓝色的圆，然后选择菜单栏中的【编辑】|【粘贴到当前位置】命令，将圆复制到“图层4”上与原来相同的位置；选择工具箱中的任意变形工具，按住Shift+Alt键，在右上角控制点处拖动，成比例从中心点放大，并将笔触颜色改为绿色，如图8-35所示。

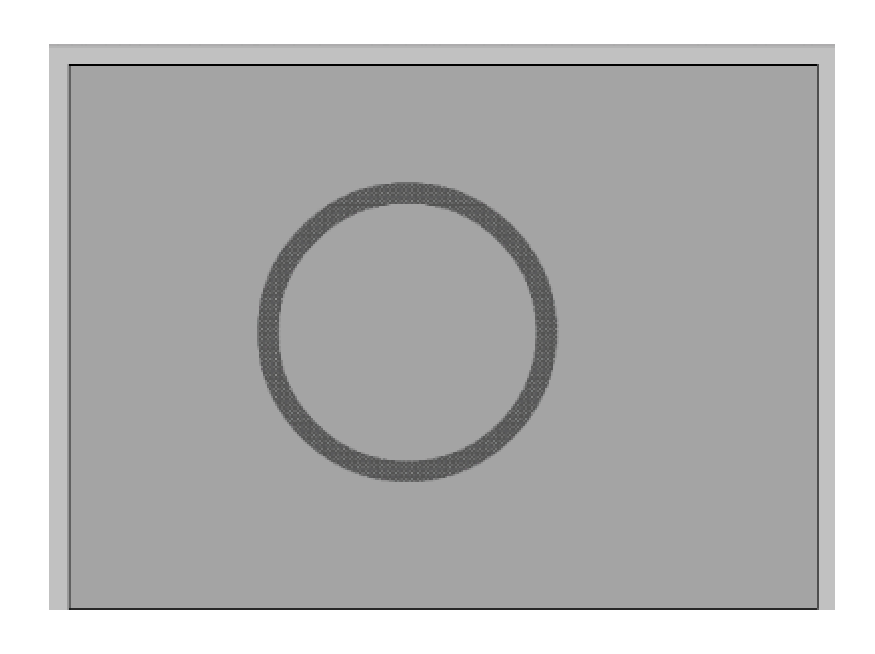


图8-34　复制正圆并修改为红色

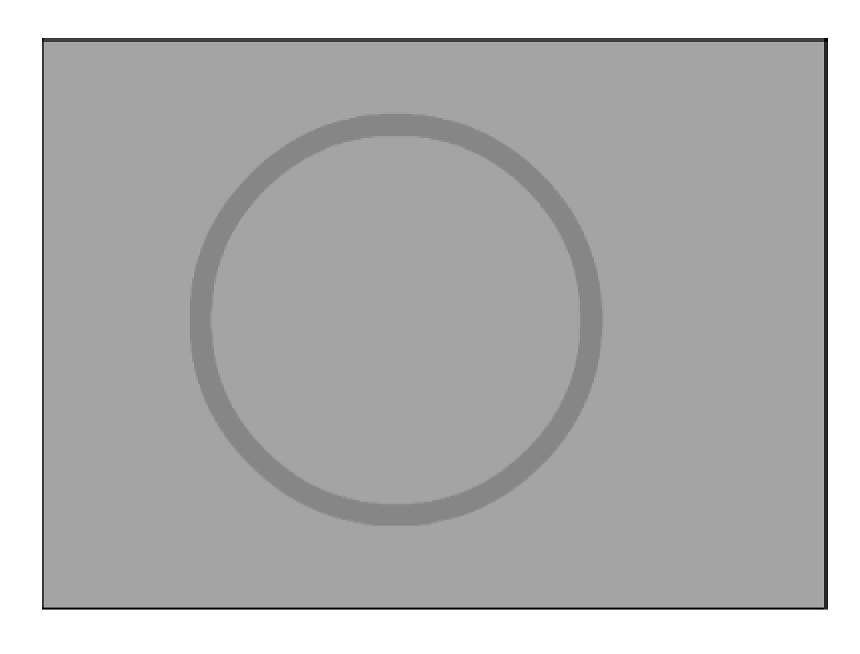


图8-35　复制正圆并修改为绿色

（6）选中“图层2”中的蓝环，在第2帧插入关键帧，并用橡皮擦工具将蓝环的某一位置擦掉一块；在第3帧插入关键帧，并用橡皮擦工具接着上一次擦除的位置继续擦除；在第4帧插入关键帧，重复擦除操作，如图8-36所示，直到将蓝环全部擦除。按住Shift键将“图层2”中的所有帧选中，然后右击，在快捷菜单中选择【翻转帧】命令。

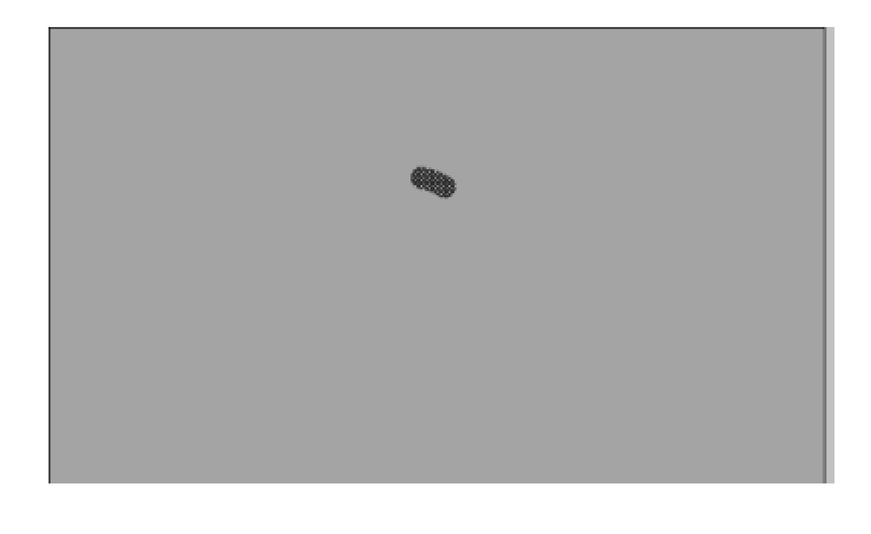


图8-36　修改关键帧

(7) 选中“图层 3”中的红环，按第 6 步的操作将红环全部擦除。

(8) 选中“图层 4”中的绿环，按第 6 步的操作将绿环全部擦除。【时间轴】面板如图 8-37 所示。

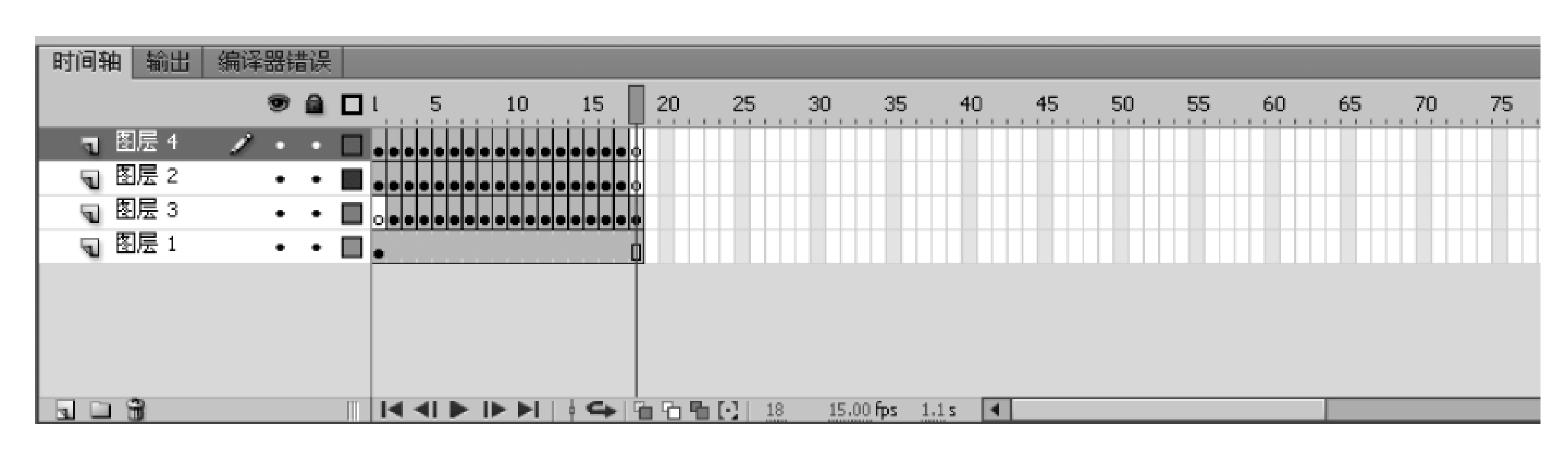


图 8-37 【时间轴】面板

(9) 新建“图层 5”，选择菜单栏中的【文件】|【导入】|【导入到舞台】命令，选择“箭头”图片素材将其导入到舞台；选中箭头，按 F8 键在【转换元件】对话框中将其转换为影片剪辑元件，如图 8-38 所示。

(10) 右击“图层 5”，在 1～18 帧中右击任意一帧，在弹出的快捷菜单中选择【创建补间动画】命令，在“图层 5”的帧中间将出现蓝色底纹；在动画的最后一帧插入关键帧，再次右击最后一个关键帧，在快捷菜单中选择【插入关键帧】|【旋转】命令，并使用 3D 旋转工具拖动鼠标设置旋转效果，如图 8-39 所示。

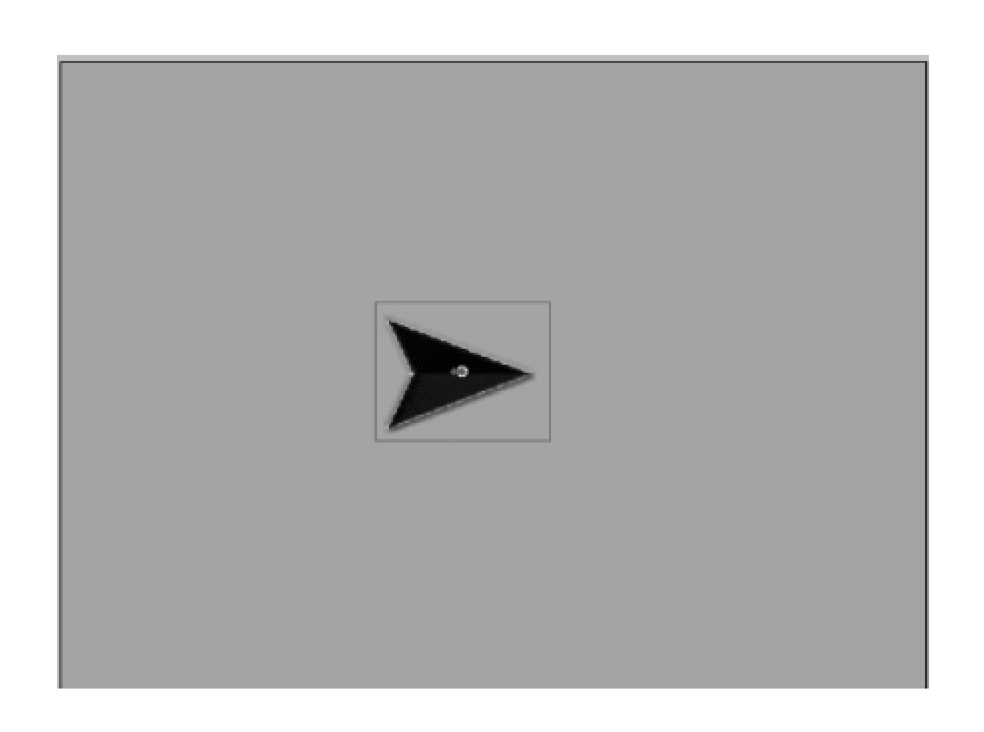


图 8-38 导入图片“箭头”并转换

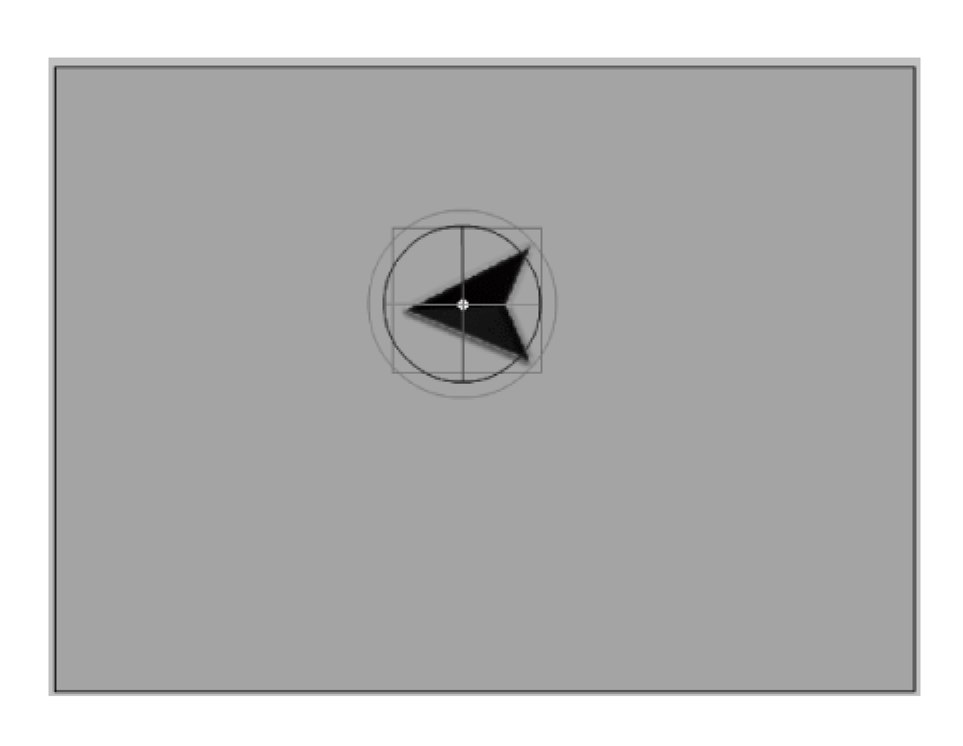


图 8-39 旋转“箭头”图片

(11) 保存，预览，最终时间轴效果如图 8-40 所示。

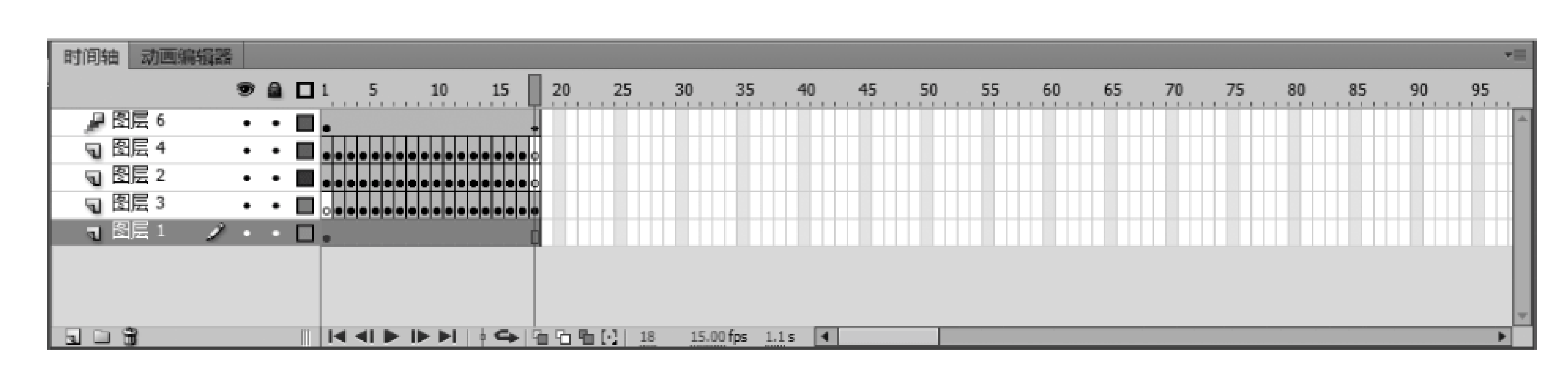


图 8-40 最终时间轴效果

第9章 高级动画的制作

本章学习目标：

✍了解引导层、遮罩层的概念。

✍了解 ActionScript 3.0 的基础知识。

✍掌握引导层、遮罩层动画的制作。

✍掌握交互动画的制作。

9.1 引导层动画

基本的动画补间动画只能使对象做直线方向的运动，如果要设计曲线运动的动画，就必须不断地设置关键帧，指定运动的路线，因此为了简化操作，Flash 中提供了引导层，让一个对象可以沿着固定的运动路径运动。

9.1.1 引导层动画的概念

引导层是 Flash 中的一个比较特殊的图层，用户可以在运动引导层中绘制对象的运动路径，然后将一个或多个图层链接到该图层中，让一个或多个对象沿同一条路径运动。在引导层中的所有内容都只能作为运动的参考，在播放时不会出现。

9.1.2 创建引导层动画

一个引导层动画由“引导层”和“被引导层”组成，其“引导层”只能有一个，而“被引导层”可以有多个。在制作引导层动画时需要创建一个“引导层”，在“引导层”上绘制对象运动的线条，然后将“被引导层”上的对象吸附到线条上，从而创建引导层动画。其方法如下：

在【时间轴】面板中选择要添加“引导层”的图层，然后右击，在弹出的快捷菜单中选择【添加传统运动引导层】命令，就可以添加运动引导层了，引导层动画的时间轴如图 9-1 所示。

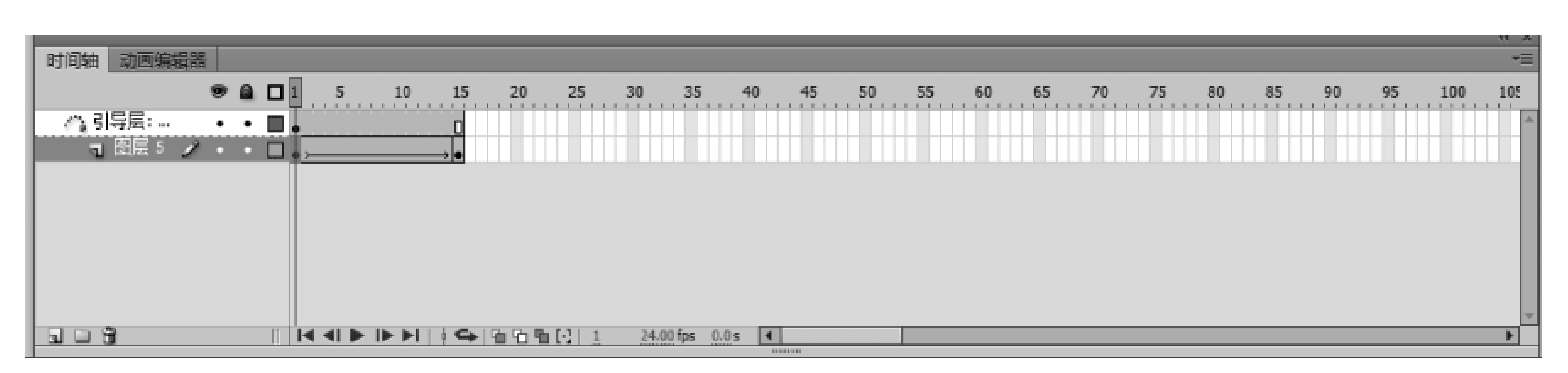


图 9-1　引导层动画的时间轴

9.1.3 引导层动画的参数设置

在默认情况下，对象在沿路径运动的过程中，不管曲线的轨迹往哪个方向，对象本身的角度是不变的，如果需要让对象按照曲线的方向来调整运动的角度，可以设置引导动画的参数。在创建引导层动画后，单击被引导图层的第 1 帧，打开【属性】面板(动画过渡【属性】面板)，其中，勾选【贴紧】复选框，可以使被引导层上对象的中心点自动吸附到路径上；勾选【调整到路径】复选框，可以使被引导层上的对象按照引导线的方向改变自己的角度，如图 9-2 所示。

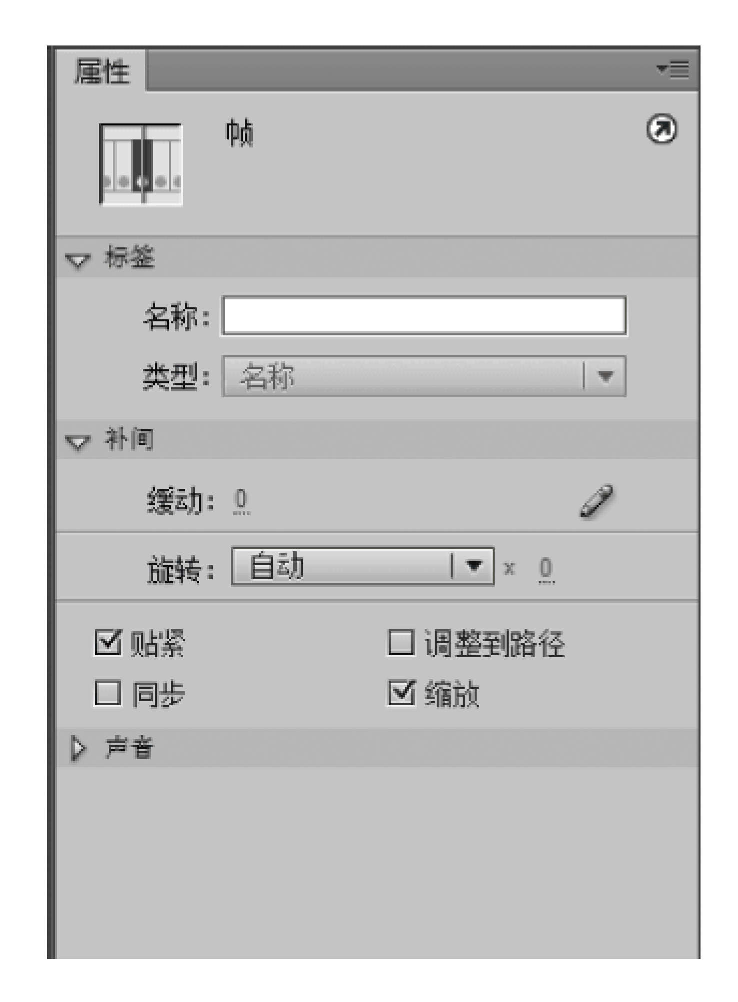


图 9-2 动画过渡【属性】面板

9.1.4 取消引导层动画

在【时间轴】面板上选择要解除引导的“引导层”，然后右击，在弹出的快捷菜单中选择【引导层】命令，此时“引导层”的图标变为普通图层图标。

9.2 遮罩层动画

遮罩动画是通过遮罩层来实现的，通过遮罩层动画可以很容易地实现水波、百叶窗等特效。

9.2.1 遮罩层的概念

在现实生活中可以看到很多类似遮罩的情况，例如通过望远镜只能看到一部分景物，在 Flash 中可以通过遮罩层来定义其下面图层中对象的可见区域。遮罩层是一种很特殊的图层，可以将下面的图层屏蔽，而下方的被遮罩层只能显示上方遮罩层图形区域大小的内容。

9.2.2 创建遮罩层动画

一个遮罩层动画由最上层的“遮罩层”和紧邻下方的“被遮罩层”组成，其中“遮罩层”只能有一个，而“被遮罩层”可以有多个。需要注意的是，“遮罩层”只是用来限定一个范围，除了线条外，可以使用任何类型的对象。

在【时间轴】面板中选择要设置为“遮罩层”的图层，然后右击，在弹出的快捷菜单中选择【遮罩层】命令，就可以添加遮罩了。应用了遮罩层动画的时间轴如图 9-3 所示。

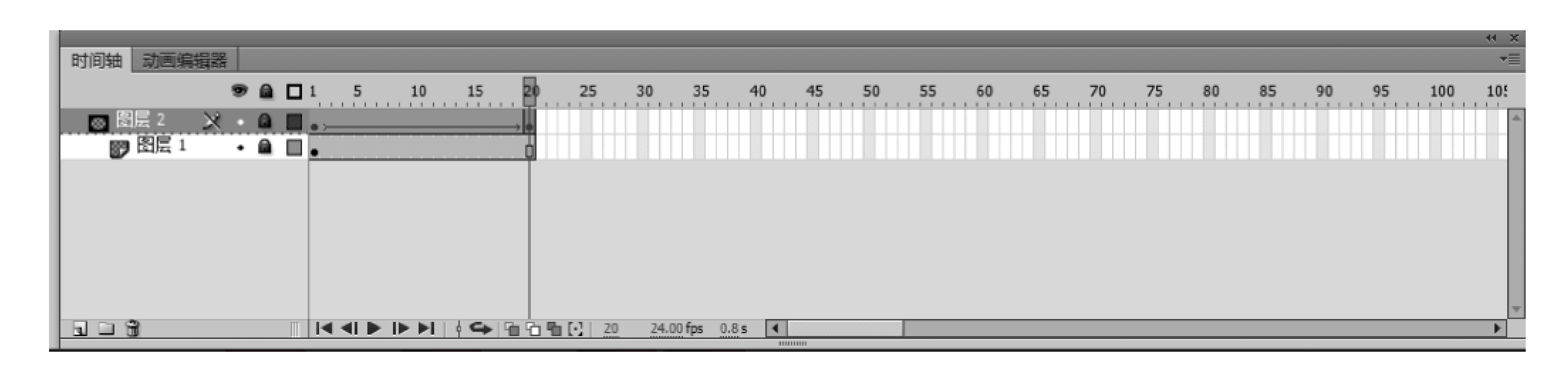


图 9-3　应用了遮罩层动画的时间轴

9.2.3 取消遮罩层

在【时间轴】面板上选择“遮罩层”，然后右击，在弹出的快捷菜单中选择【遮罩层】命令，此时“遮罩层”和“被遮罩层”的图标都变为普通层。

9.3 交互式动画

9.3.1 ActionScript 3.0 简介

ActionScript 是由 Flash Player 中的 ActionScript 虚拟机（AVM）来执行的。ActionScript 代码通常被编译器编译成“字节码格式”（一种由计算机编写且能够为计算机所理解的编程语言），如 Adobe Flash CS6 Professional 或 Adobe Flex Builder 的内置编译器，或 Adobe Flex SDK 和 Flex Data Services 中提供的编译器。将字节码嵌入 SWF 文件中，SWF 文件由运行时环境 Flash Player 执行。

AS3.0 的脚本编写功能超越了 AS 的早期版本，它旨在方便创建拥有大型数据集和面向对象的可重用代码库的高度复杂的应用程序，虽然 AS 3.0 对于在 Adobe Flash Player 9 中运行的内容并不是必需的，但它使用新型的虚拟机 AVM2 实现了性能的改善。AS3.0 代码的执行速度比旧式的 AS 代码快了 10 倍。

AS3.0 提供了可靠的编程模型，具备面向对象编程基本知识的开发人员对此模型会感受到似曾相识，它的一些主要功能如下：

- 一个新增的 ActionScript 虚拟机，称为 AVM2，它使用全新的字节码指令集，可以使性能显著提高。

- 一个更为先进的编译器代码库,它更为严格地遵循 ECMAScript(ECMA262)标准,并且相对于早期的编译器版本可执行更深入的优化。
- 一个扩展并改进的应用程序编程接口(API),拥有对对象的低级控制和真正意义上的面向对象模型。
- 一种基于即将发布的 ECMAScript(ECMA-262)第 4 版草案语言规范的核心语言。
- 一个基于文档对象模型(DOM)第 3 级事件规范的事件模型。

9.3.2 常用的数据类型

在 ActionScript 3.0 中经常要用到的数据类型为所有的基元数据类型(Boolean、int、uint、Number、String)和两种复杂的数据类型(Array、Object)。

1. 字符串(String)

字符串是一对用双引号括起来的字母、数字、特殊字符的组合,如"Hello World!"。引号内的字符是区分大小写的,字符串型数据能进行字符串连接运算。

2. 布尔值(Boolean)

布尔值是用来表示真假的数据类型,它只有 true(真)和 false(假)两个值。如果只声明一个布尔类型的变量,没有给它赋值,默认值是 false。

3. 数字(int、uint、Number)

在 ActionScript 3.0 中表示数字值的数据类型很少,只有 3 种,即 int、uint、Number。其中,int 和 uint 是整型数值,专门用来处理整数,而 Number 是用来处理浮点数的。

4. 数组(Array)

数组是可以作为数据的容器,它能包含更多的数据,每一个元素(数据中的一部分)都被依附于一个索引。

数组的声明方式有几种,如表 9-1 所示。

表 9-1 数组的声明方式

数　　组	声　　明
var a :Array	声明一个数组变量 *a*,但还没有告诉 *a* 的引用指向谁,trace 得到 null
var b: Array=[ ]	直接声明一个空数组变量 *b*,trace 得到空白的显示,但不再是 null 了
var c: Array=new Array( );	效果同 *b* 的方法
var d: Array=[1,2,3,4];	直接使用[]操作符,建立一个含有整数 1、2、3、4 的数组
var e: Array=new Array(1,2,3,4);	使用 Array 类进行和 *d* 同样的操作
var f: Array=new Array(5);	声明一个长度为 5 的空数组,此时每个数组元素都为空

访问数组中的元素只需要知道元素的位置,就可以使用数组运算符"[]"来访问它。例如 d[0]、d[1],索引 0 代表第一个元素,第二个元素的索引值为 1,其余依此类推。

5. 对象(Object)

对象数据类型包含大量复杂的信息，是 ActionScript 所有数据结构的基石，是面向对象编程思想的载体。对象是属性的集合，每个属性都有名字和值；属性值可以是任何一种 Flash 支持的数据类型，甚至是对象类型；对象及其属性的设定需要使用“.”操作符。

点语法“.”用于设置对象或影片剪辑的属性和方法，它也用于标识指向影片剪辑或变量的目标路径。一个点语法表达式以对象或影片剪辑的名字开始，后面跟着一个点，以属性、方法或者变量结束，在这两组之间可以插入路径。

例如，_x 表示一个影片剪辑实例在 X 轴的位置，而 newMC._x 用来指出影片剪辑实例 newMC 的 X 轴的位置。

9.3.3　语法规则

动作脚本拥有自己的一套语法规则和标点符号。

1. 运算符

在动作脚本中，点“.”用于表示与对象或影片剪辑相关联的属性或方法，也可用于标识影片剪辑或变量的目标路径。点运算符表达式以影片或对象的名称开始，中间为点运算符，最后是要指定的元素。

例如，_x 影片剪辑属性指示影片剪辑在舞台上的 X 轴的位置，表达式 aaMC_x 引用影片剪辑实例 aaMC 的_x 属性。

无论是表达对象的方法还是影片剪辑的方法，均遵循同样的模式。例如，aa_MC 影片剪辑实例的 play()方法在 aa_MC 的时间轴中移动播放头，如下面的语句所示：

```
aa_MC.play();
```

点语法还使用两个特殊别名_root 和_parent，别名_root 指主时间轴，可以使用_root 别名创建一个绝对目标路径。例如，下面的语句调用主时间轴上的影片剪辑 function 中的 buildGrameBoard()：

```
_root.function. buildGrameBoard();
```

可以使用别名_parent 引用当前对象嵌入到的影片剪辑，也可以对使用_parent 创建相对目标路径。例如，如果影片剪辑 aa_MC 嵌入影片剪辑 bb_MC 的内部，则实例 aa_MC 的如下语句会指示 bb_MC 停止：

```
parent.bb_mc.stop();
```

2. 界定符

(1) 大括号：动作脚本中的语句可被大括号包括起来组成语句块。
例如：

```
//事件处理函数
```

```
public Function myDate(){
Var myData:Date = new Date( );
currentMonth = myDate.getMonth( );
}
```

(2) 分号：动作脚本中的语句可以由一个分号结尾。如果在结尾处省略分号，Flash 仍然可以成功编译脚本。

例如：

```
var colum = passedDate.getDay( );
var row = 0;
```

(3) 圆括号：在定义函数时，任何参数定义都必须放在一对圆括号内。

例如：

```
function myFunction(name,age,reader){
}
```

在调用函数时，需要被传递的参数也必须放在一对圆括号内，例如：

```
myFunction("Steve",10,true);
```

可以使用圆括号改变动作脚本的优先顺序或增强程序的易读性。

3. 区分大小写

在区分大小写的编程语言中，仅大小写不同的变量名(book 和 Book)被视为互不相同，ActionScript 3.0 中的标识符区分大小写，例如下面两条动作语句是不同的：

```
cat.hilite = true;
CAT.hilite = true;
```

对于关键字、类名、变量、方法名等，要严格区分大小写。如果关键字大小写出现错误，在编写程序时会有错误信息提示；如果采用了彩色语法模式，那么正确的关键字将以深蓝色显示。

4. 注释

在【动作】面板中，使用注释语句可以在一个帧或者按钮的脚本中添加说明，有利于增加程序的易读性。注释语句以双斜线“//”开始，斜线显示为灰色；注释内容可以不考虑长度和语法，注释语句不会影响 Flash 动画输出时的文件量。例如：

```
public Function myDate( ){
//创建新的 Date 对象
var my Date;
Date = new Date();
currentMonth = myDate.getMonth();
//将月份数转换为月份名称
monthName = calcMonth(currentMonth);
year = myDate.getFullYear();
currentDate = myDate.getDate();
}
```

5. 关键字

动作脚本保留了一些单词用于该语言中的特定用途，因此不能将它们用作变量、函数或标签的名称；如果在编写程序的过程中使用了关键字，动作编辑框中的关键字会以蓝色显示。为了避免冲突，在命名时可以展开动作工具箱中的 Index 域，检查是否使用了已定义的名字。

6. 常量

常量中的值永远不会改变，所有的常量都可以在【动作】面板的工具箱和动作脚本字典中找到。

9.3.4 变量及运算符、表达式

1. 变量的声明

在 ActionScript 3.0 中声明变量的格式如下：

```
var 变量名:数据类型;
var 变量名:数据类型 = 值;
```

var 是一个关键字，用来声明变量。变量的数据类型写在冒号后；其次，如果要赋值，值的数据类型必须和变量的数据类型一致。

ActionScript 3.0 代码每行的结尾若不加上分号不会导致报错，但是为了使代码标准化，应当加上。另外，某些 ActionScript IDE(开发环境)会对句尾不加分号的情况报错。

```
//错误的例子
i;                                  //没有加 var 关键字,即没有声明变量,出错
i = 3;                              //没有加 var 关键字,出错
var j:int = "String Value";
//声明变量的数据类型为 int,却赋予了一个字符串的值,出错
//正确的例子
var i : int;                        //声明了一个 int 型变量,但没有赋值,只好使用默认值
var k : int = 100;                  //声明了一个 int 型变量,并赋值 100
var h;                              //声明变量 h,但未指定类型,默认值为 untyped
var g : * ;                         //声明变量 g,效果同上行
```

2. 变量的命名

命名规则不仅仅是为了让编写的代码符合语法，更重要的是增强自己代码的可读性，要做到让自己能看清楚，让别人能看明白。

(1) 尽量使用有含义的英文单词作为变量名(例如，一个变量名为 address，这个变量存储的应该是地址)。

(2) 变量名采用骆驼式命名法(骆驼式命名法是指混合使用大小写字母来构成变量的名字。若首字母小写，第二个词的首字母大写)。

(3) 变量名不能是 ActionScript 3.0 内部定义的保留字和关键字，ActionScript 3.0 保

留了一些专门用于本语言中不能用于变量、函数名等的定义，常用的关键字有 break、for、new、var、continue、function、return、void、delete、if、this、while、typeof、with、set、get、try、true、false、class、else、each、include、null、is、in、const。

3. 运算符、表达式

编程语言必须要清楚地描述如何进行数据运算，因此必须通过某种表达方式告诉计算机进行什么样的数据运算。

(1) 赋值运算符：=。

(2) 算术运算：+、-、*、/、%(模)。

(3) 算术赋值运算符：+=、-=、*=、/=、%=，等价关系见表 9-2。

表 9-2 算术赋值运算符的等价表达式

运 算 符	运算符举例	等价表达式
+=	a+=b	a=a+b
-=	a-=b	a=a-b
*=	a*=b	a=a*b
/=	a/=b	a=a/b
%=	a%=b	a=a%b

(4) 关系运算符：==、!=、>=、<=、>、<。

这些操作比较简单，这里不再多讲，注意其返回值为布尔值。

(5) 逻辑运算符：&&、||、!。

逻辑运算符比较好理解，只包括 3 个运算符。

- 逻辑与(&&)：当两边表达式都为 true 时，返回值为 true。
- 逻辑或(||)：当两边表达式有一个值为 true 时，返回值为 true。
- 逻辑非(!)：只有一个运算对象，在右边，其布尔值取反。

9.3.5 流程控制语句

程序流程是程序执行的方向，是语句执行的先后顺序。在 ActionScript 3.0 中有 3 种结构的控制流程，即顺序、分支和循环。任何复杂的程序都可以由这 3 种结构组成，这 3 种结构相互包含，嵌套使用。

1. 顺序结构

顺序结构是程序最基本、最简单的结构，在分支结构和循环结构中也包含顺序结构。在顺序结构中，程序按照语句出现的先后顺序依次执行，直到到达最后的语句。顺序结构的示意图如图 9-4 所示。

2. 分支结构

分支结构是程序依据预先设定的条件成立与否决定执行哪个分支，分支结构由 if 条件语句实现，分支结构也称为条件结构。分支结构的示意图如图 9-5 所示。

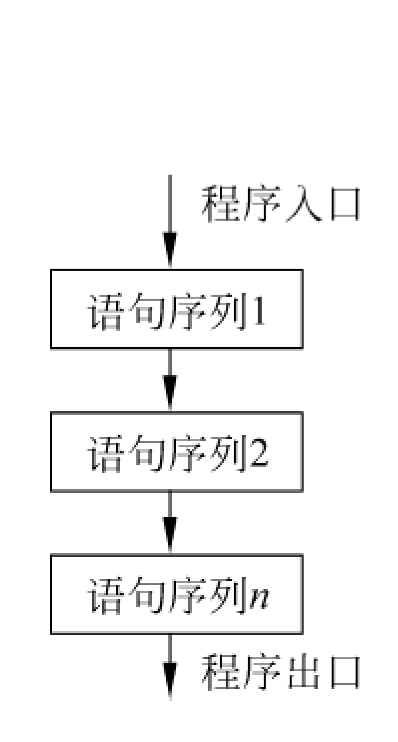


图 9-4　顺序结构的示意图

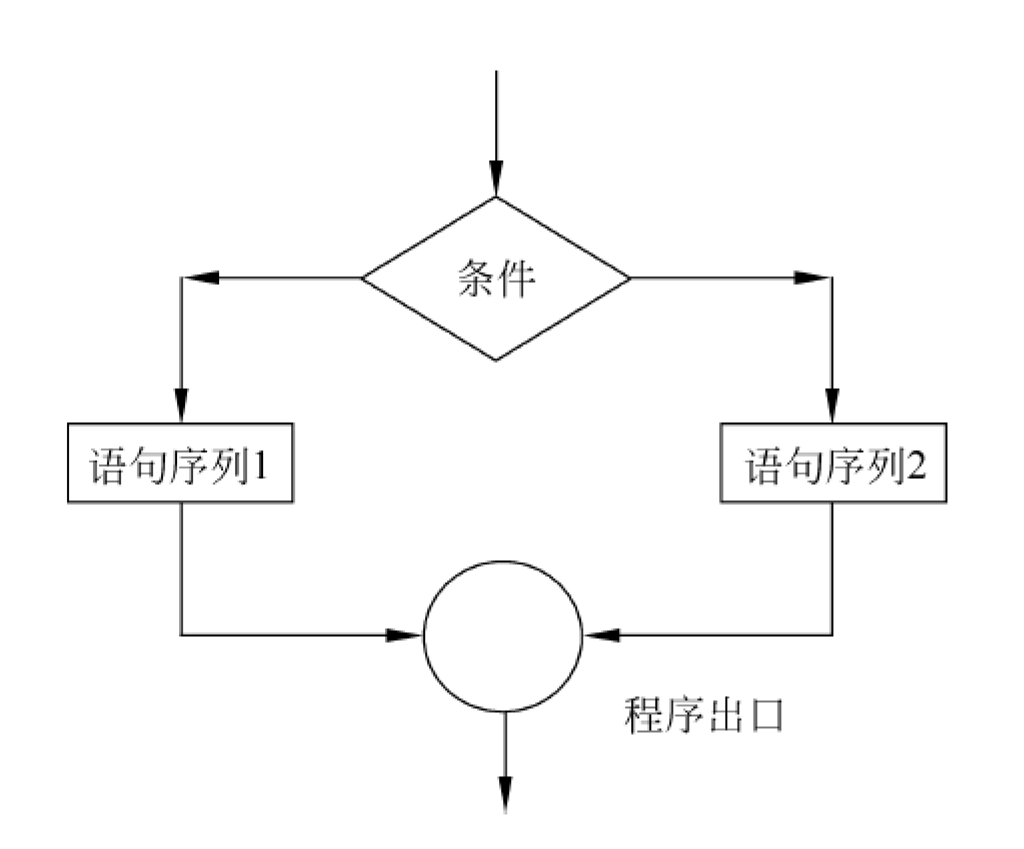


图 9-5　分支结构的示意图

整个 if 条件语句的语法格式如下：

```
if(条件 1)
{
    代码段 1
}else if(条件 2)
    {
        代码段 2
    }
    …
      else
      {
          代码段 n
      }
```

在 if 条件结构中，else 选项可要可不要。当程序执行到 if 语句时，首先判断条件 1 是否成立，如果成立，则执行代码段 1，然后退出条件结构，执行条件结构后面的语句；如果条件 1 不成立，则依次判断条件 2 是否成立，如果成立，执行代码段 2，退出条件结构；如果不成立，继续判断设定的其他条件；当设定的所有条件都不成立时，执行 else 后面的代码段 n。

条件语句可以嵌套使用，也就是说上述代码段还可以再包含条件语句，此项功能使得用户能设计出条件更复杂的程序，只要用户愿意，完全可以按照自己的要求来选择条件嵌套的层数。

3. 循环结构

循环结构是程序依据预先设定的条件是否成立反复执行相同的代码段，当条件不成立时退出循环。循环结构主要是用 for 语句和 while 语句实现的。循环结构的示意图如图 9-6 所示。

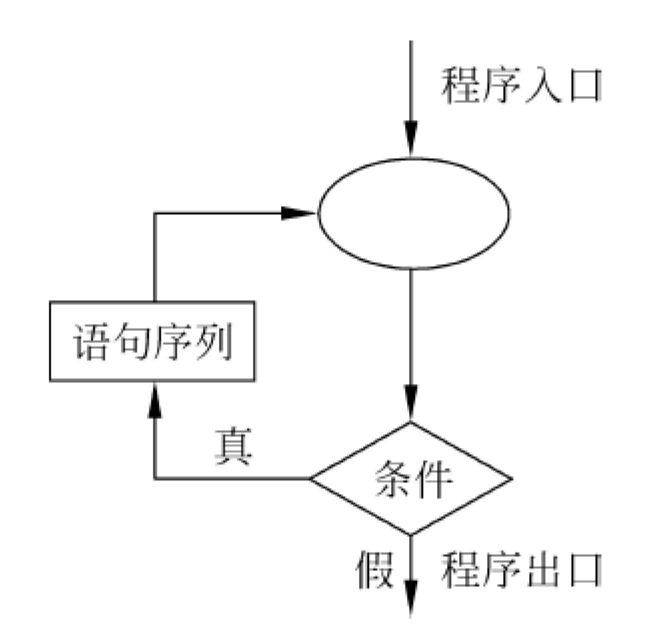


图 9-6　循环结构的示意图

1）for 循环

for 循环比较灵活，应用最为广泛，其循环结构如下：

```
for(初始化;循环条件;步进)
```

```
{
    循环体
}
```

2）while 循环

while 的中文意思是“当……的时候”，所以 while 循环表示当条件满足表达式的时候再执行循环体。它的结构如下：

```
while(循环条件)
{
    循环体
}
```

无论是 for 循环还是 while 循环，在设计时都要注意死循环。所谓死循环就是用来作为判断的条件始终成立，程序将不停地循环，不能正常退出，导致异常情况发生。循环变量的边界值是最容易出错的，如循环变量的初值或终值设定不对，使得程序多循环或少循环，从而发生不可预料的结果。对于以上两点大家在编程时要特别小心。

另外，还有其他的循环语句，如 for...in、for each...in 以及 break、continue、swith 等，由于初学者用得比较少，这里不再赘述，对编程感兴趣的同学可以参考网上的相关资料。

9.3.6 利用【动作】面板添加动作

1.【动作】面板

若要创建嵌入到 FLA 文件中的脚本，可以直接将 ActionScript 输入到【动作】面板中，选择菜单栏中的【窗口】|【动作】命令或者按 F9 键可以打开【动作】面板。

【动作】面板由 4 个部分构成，即动作工具箱(按类别对 ActionScript 元素进行分组)、脚本导航器(可以快速地在 Flash 文档中的脚本间导航)、脚本窗口(可以在其中输入 ActionScript 代码)和面板选项菜单。

1）动作工具箱

如果将 ActionScript 元素插入到脚本窗口中，可以双击该元素，或直接将它拖动到脚本窗口。动作工具箱将项目进行了分类，并且提供了按字母顺序排列的索引。

2）脚本导航器

单击脚本导航器中的某一项目，与该项目关联的脚本将显示在脚本窗口中，并且播放头将移动到时间轴上的相应位置。双击脚本导航器中的某一项目可固定脚本(将其锁定在当前位置)。

3）脚本窗口

如果同时打开多个外部文件，文件名将显示在沿脚本窗口顶部排列的选项卡上。

在脚本窗口中可以使用以下功能：“添加”(＋)菜单(类似动作工具箱)、查找和替换、语法检查、语法着色、自动套用格式、代码提示、代码注释、代码折叠、调试选项(仅限 ActionScript 文件)和自动换行。使用脚本窗口可以显示行号和隐藏字符。

脚本窗口不具有代码帮助功能，例如脚本导航器、脚本助手模式和行为，这些功能只有在创建 Flash 文件时才有用，对创建外部脚本文件没有帮助。

4）面板选项菜单

该菜单中包含适用于【动作】面板的命令和首选参数，例如可以设置行号和自动换行、访问 ActionScript 首选参数以及导入或导出脚本。

2. 给帧添加动作

给关键帧指定一个动作，以使电影在到达那一帧时做一些事情，也是控制时间轴的动作。

常见的就是对时间轴的控制，对时间轴的控制一般使用 play()（播放）、gotoAndPlay()（跳转到某一帧播放）、stop()（停止）、gotoAndStop()（跳转到某一帧停止）语句。

3. 给按钮添加动作

为按钮实例添加动作，可以使用户在按下鼠标或者在鼠标经过按钮等其他事件触发时执行动作，给一个按钮实例添加动作不会影响其他按钮的动作。

当给一个按钮添加动作时，可以指定触发动作的鼠标事件，也可以指定一个触发动作的键盘中的某一键。

当给按钮设置动作时，必须把该动作嵌套在鼠标事件 on(mouse event)处理程序中，并指定触发该动作的鼠标或键盘事件，其基本表达式如下：

```
on(鼠标事件){
    控制语句;
}
```

常见的控制语句有 play()、stop()、gotoAndPlay()(跳转到某一帧或者某场景的某一帧播放)、gotoAndStop(跳转到某一帧或者某场景的某一帧停止)。

4. 给影片剪辑元件添加动作

通过为影片剪辑指定动作，可以在影片剪辑元件加载或接收到数据时让影片执行动作。和为按钮添加动作一样，必须将动作指定给影片剪辑元件的一个实例。影片剪辑元件的鼠标事件如表 9-3 所示。

表 9-3 影片剪辑元件的鼠标事件一览表

鼠标事件	触发条件
load	当影片片段第一次加载到时间轴时会触发本事件一次
unLoad	当影片片段被删除时会触发本事件一次
enterFrame	影片片段加载时间轴时，不论是放映还是停止状态，都会不断触发本事件，所以只要此片段被加载，此事件会一直不断地执行，直到影片片段被删为止
mouseDown	当鼠标左键被按下时会触发本事件一次
mouseUp	当按下的鼠标左键被放开时会触发本事件一次
mouseMove	只要在场景中移动鼠标就会不断触发本事件
keyDown	当键盘被按下时会触发本事件
keyUp	当已按下的键盘被松开时会触发本事件一次
data	当在 loadVariables() 或 loadMovie() 动作中接收数据时启动此动作。当与 loadVariables()动作一起指定时，data 事件只在加载最后一个变量时发生一次；当与 loadMovie()动作一起指定，获取数据的每一部分，data 事件重复发生

当为影片剪辑元件指定动作时，必须将动作嵌套在 onClipEvent 处理函数中，并指定触发该动作的剪辑事件，表达式如下：

```
onClipEvent(鼠标事件){
    控制语句;
}
```

用于控制影片剪辑元件的动作主要有 startDrag(开始拖动)、stopDrag(停止拖动)、duplicateMovieClip(复制影片剪辑元件)、removeMovieClip(删除影片剪辑元件)、tellTarget(指定影片剪辑元件)、setProperty(设置属性)等。

9.4 综合应用

9.4.1 星空的制作

制作一个类似于星空的动画，球沿着给定的圆环运动，由小变大，其操作步骤如下：

(1) 选择菜单栏中的【文件】|【新建】命令，建立一个 500 像素×500 像素大小的 Flash 文档。

(2) 选择菜单栏中的【文件】|【导入】|【导入到库】命令，将"路径"素材图片和"球"素材图片导入到库中。

(3) 单击"图层 1"的第 1 帧，将"路径"素材拖至舞台上，并调整好位置，如图 9-7 所示。

(4) 创建"图层 2"，将"球"素材拖至舞台上，然后选中"球"，选择菜单栏中的【修改】|【转换为元件】命令，将小球转换为图形元件，如图 9-8 所示。

(5) 选中"图层 2"，右击【图层】面板，在弹出的快捷菜单中选择【添加传统运动引导层】命令，为"图层 2"添加运动引导层。然后选择铅笔工具，在引导层上绘制螺旋线，并在中间部分用橡皮擦擦除一小块，如图 9-9 所示。

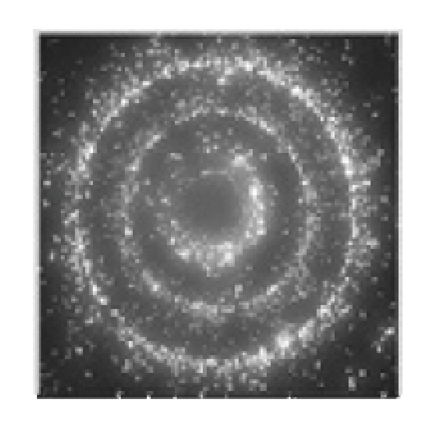


图 9-7 编辑图层 1

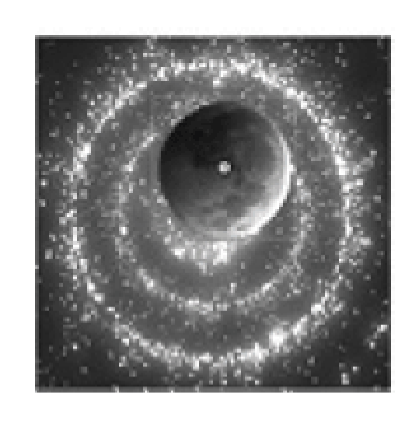


图 9-8 编辑图层 2



图 9-9 编辑图层 2 的引导层

(6) 单击"图层 2"，在第 1 帧拖动小球至线最里面的顶点；在第 21 帧插入关键帧，将小球拖至橡皮擦擦除的地方的顶点；在第 22 帧插入关键帧，将小球拖至橡皮擦擦除的地方的另一个顶点；在第 45 帧插入关键帧，拖动小球到线最外面的顶点；用鼠标在 1～21 帧之间的任意帧右击，在弹出的快捷菜单中选择【创建传统补间】命令，同样，在 22～45 帧之间也创建传统补间动画，【时间轴】面板如图 9-10 所示。

(7) 单击小球所在图层的第 1 帧，选择小球，使用任意变形工具将小球调小。同样，在第 21 帧将小球调大一些，在第 45 帧将小球调得更大一些，如图 9-11 所示。

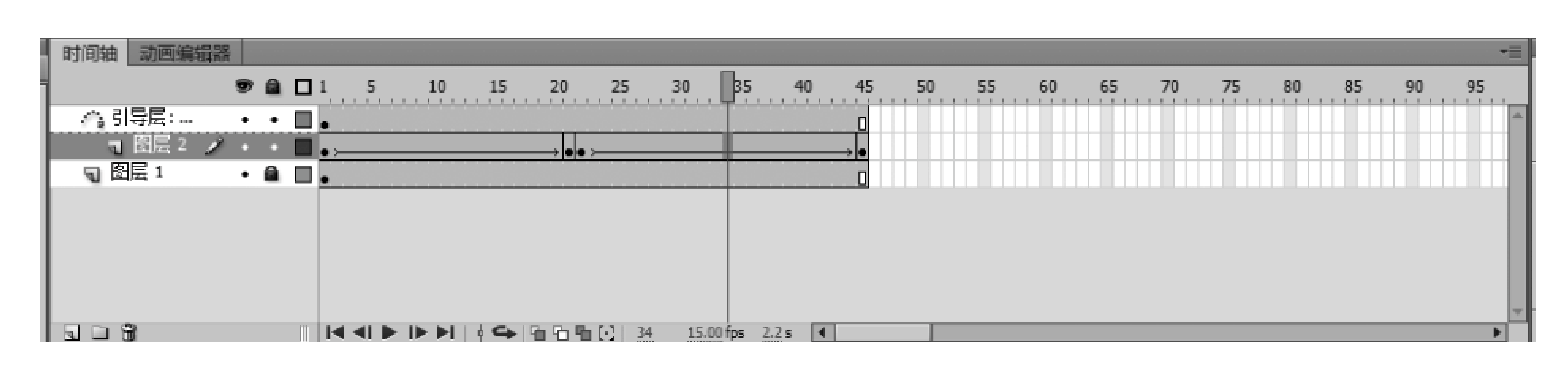


图 9-10　【时间轴】面板

(8) 保存,预览,最终效果如图 9-12 所示。



图 9-11　关键帧的小球修改

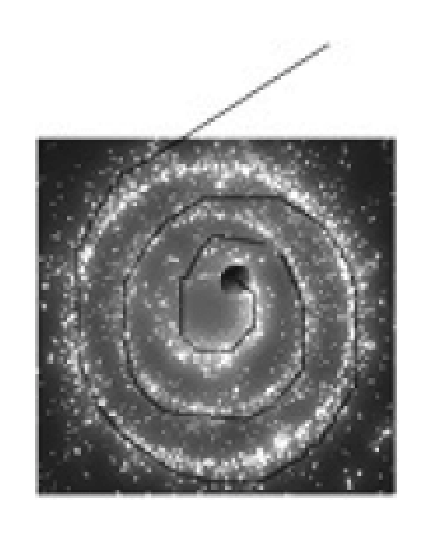


图 9-12　星空案例的最终效果

9.4.2　Banner 的制作

制作一个 Banner,曲线颜色渐变,文字一个一个出现,其操作步骤如下:

(1) 选择菜单栏中的【文件】|【新建】命令,创建一个 480 像素×60 像素大小的 Flash 文档,如图 9-13 所示。



图 9-13　新建文件

(2) 选择矩形工具,绘制一个 480 像素×60 像素大小的矩形,并用浅蓝到深蓝线性渐变填充,然后使用渐变变形工具调整渐变的方向,如图 9-14 所示。



图 9-14　绘制并编辑矩形

（3）单击【时间轴】面板中的【新建图层】按钮新建“图层 2”，然后选择钢笔工具，选中“图层 2”，在舞台上单击 3 次得到一个线段，并利用转换锚点工具将尖角处转换为圆角，调整线段，得到一个弧线，再选择油漆桶工具，从白色到黑色线性渐变填充，如图 9-15 所示。



图 9-15 绘制并编辑弧线

（4）单击时间轴的第 10 帧，然后右击，在快捷菜单中选择【插入关键帧】命令，在【颜色】面板中调整线性填充色带中的色标位置；同样，在第 20 帧插入关键帧，调整线性填充；在 1～10 帧、11～20 帧之间创建补间形状，如图 9-16 所示。



图 9-16 弧线颜色渐变

（5）单击【时间轴】面板中的【新建图层】按钮新建“图层 3”，然后选择工具箱中的文本工具，在舞台上单击输入文字“hello world!!!”；选中文字，在【属性】面板中设置字体为 Arial、大小为 30 点、颜色为白色、字母间距为 5，如图 9-17 所示。

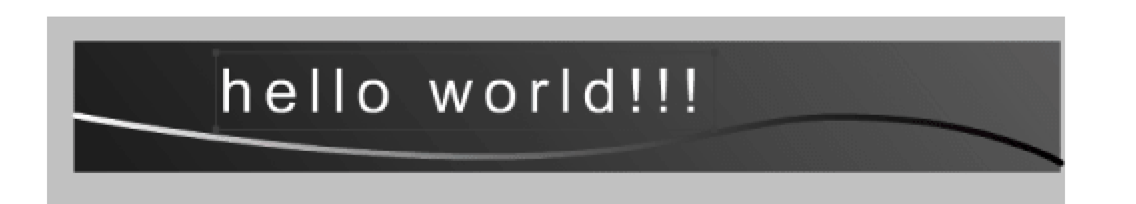


图 9-17 编辑文字

（6）用同样方法新建“图层 4”，然后选择第 1 帧，使用矩形工具在文字最左侧绘制一个小矩形，以任意颜色填充，无边框；在第 20 帧插入关键帧，使用任意变形工具拖动小矩形使其变大，覆盖文字，如图 9-18 所示。

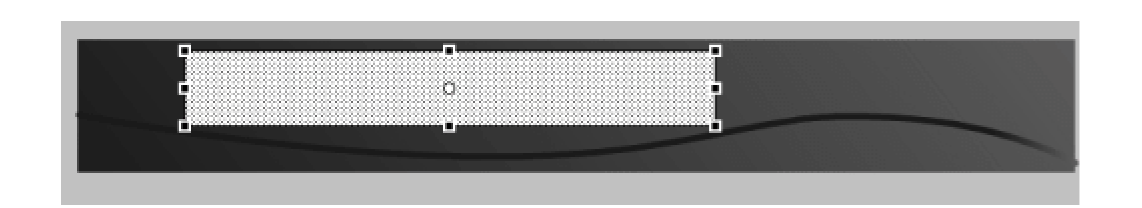


图 9-18 编辑图层 4

（7）选中最上面的“图层 4”，然后右击，在快捷菜单中选择【遮罩层】命令，最终效果如图 9-19 所示。【时间轴】面板如图 9-20 所示。



图 9-19 Banner 的最终效果

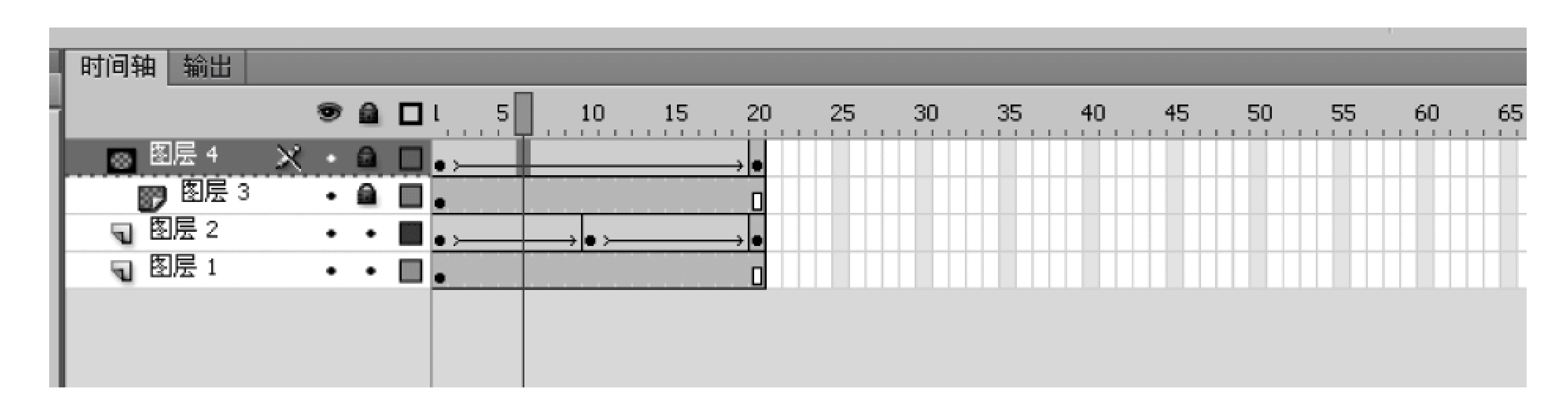


图 9-20　【时间轴】面板

9.4.3　动作按钮的应用

制作动作按钮，用按钮控制影片剪辑的播放与停止，其操作步骤如下：

(1) 将 9.4.1 节中动画的所有帧全部选中，选择菜单栏中的【插入】|【新建元件】命令，在元件编辑窗口中右击，在快捷菜单中选择【粘贴帧】命令，将该动画转换为一个影片剪辑，命名为“影片”。选择菜单栏中的【文件】|【新建】命令，创建一个 480 像素×120 像素大小的 Flash 文档，保存，命名为“按钮动作”。将上述影片剪辑拖到“按钮动作”文件库中，并应用到舞台，移动到适当位置，如图 9-21 所示。



图 9-21　影片剪辑

(2) 新建“图层 2”，选择菜单栏中的【窗口】|【公共库】| button 命令，在打开的【外部库】面板中选择想要的按钮，将其拖入舞台中。如图 9-22 所示，拖入一个播放按钮、一个暂停按钮。



图 9-22　在影片剪辑中加入按钮

(3) 选择“图层 1”中的影片剪辑，在【属性】面板中将其实例名称命名为 mc。选择“图层 2”中的播放按钮，命名为 my_buttonplay，选择停止按钮命名为 my_buttonstop，【属性】面板如图 9-23 所示。



图 9-23 【属性】面板

（4）单击播放按钮，选择菜单栏中的【窗口】|【动作】命令，打开【动作】面板，选择“图层 2”的第 1 帧，在动作窗口中输入以下代码：

```
my_buttonplay.addEventListener(MouseEvent.MOUSE_DOWN,funonDown);
function funonDown(e:MouseEvent){
mc.play()
}
my_buttonstop.addEventListener(MouseEvent.MOUSE_DOWN,funonup);
function funonup(e:MouseEvent){
mc.stop()
}
```

（5）保存、预览。

第10章 HTML与CSS网页设计基础

本章学习目标：

✍了解 HTML 的概念及特点。

✍了解 CSS 的概念及特点。

✍掌握 Dreamweaver 的常用操作。

10.1 HTML 简介

HTML 是使用特殊标记来描述文档结构和表现形式的一种语言，由 W3C（World Wide Web Consortium）所制定和更新。我们可以用任何一种文本编译器来编辑 HTML 文件，因为它就是一种纯文本文件。

10.1.1 HTML 的概念

HTML（HyperText Mark-up Language）即超文本标记语言，是 WWW 的描述语言。它是用于创建可跨平台的超文本文档的一种简单标记语言，它是由 HTML 命令组成的描述性文本，不仅可以用来描述文字、图形、动画、声音、表格、链接等，同时它又通过标记（TAG）来指明网页中的文字、图像、动画等元素是如何显示的。

使用 HTML 语言编写的脚本一般被称为网页或 HTML 文档，它的文件扩展名通常为.htm 或.html。

10.1.2 HTML 的产生及特点

HTML 作为定义万维网的基本规则之一，最初由蒂姆·伯纳斯-李（Tim Berners-Lee）于 1989 年在 CERN（Conseil Europeen pour la Recherche Nucleaire）研制出来。它里面包含了大约 20 个用来标记网页的 HTML 标记，他直接借用了 SGML 的标记格式，也就是后来人们看到的 HTML 标记格式，在后期逐渐发展，不断产生新型的、功能强大且生动有趣的标记形式，到 1999 年 12 月 24 日，W3C 推荐的标准 HTML 4.01 加入了很多特定浏览器的元素和属性，但是同时也开始“清理”这个标准，把一些元素和属性标记为过时，建议人们不再使用它们。目前 HTML 的最高版本为 5.0，它与 CSS 结合得更加密切。

HTML 文档的制作不是很复杂，且功能强大，支持不同数据格式的文件嵌入，这也是 WWW 盛行的原因之一，其主要特点如下：

1. 简易性

HTML 版本升级采用超集方式，从而更加灵活、方便。

2. 可扩展性

HTML 语言的广泛应用带来了加强功能，增加标识符等要求，HTML 采取子类元素的方式，为系统的扩展带来保证。

3. 平台无关性

虽然 PC 大行其道，但使用 MAC 等其他机器的大有人在，HTML 可以被用在广泛的平台上。

10.1.3 HTML 和 XHTML

HTML 和 XHTML 的区别简单来说就是 XHMTL 是语法要求更加严格的 HTML，XHTML 可以认为是 XML 版本的 HTML。

XHTML 解决了 HTML 语言所存在的严重制约其发展的问题。HTML 发展到今天存在 3 个主要缺点：不能适应越来越多的网络设备和应用的需要，例如手机、PDA、信息家电都不能直接显示 HTML；由于 HTML 代码不规范、臃肿，浏览器需要足够智能和庞大才能正确显示 HTML；数据与表现混杂，这样的页面要改变显示就必须重新制作 HTML。于是 W3C 又制定了 XHTML 标准，XHTML 是 HTML 向 XML 过渡的一个“桥梁”。

10.2 CSS 简介

CSS 算是网页设计的一个突破，它解决了网页界面排版的难题。可以这么说，HTML 主要是定义网页的内容(Content)，而 CSS 决定这些网页内容如何显示(Layout)。

10.2.1 CSS 的概念

CSS 语言是“Cascading Style Sheets”的缩写，中文翻译为“层叠式样式表单”，它是由 W3C 协会制定并发布的一个网页排版标准，是一种用来表现 HTML(标准通用标记语言的一个应用)或 XML(标准通用标记语言的一个子集)等文件样式的计算机语言，是对 HTML 语言功能的补充，主要的用途是对网页中字体、颜色、背景、图像及其他各种元素的控制，使网页能够完全按照设计者的要求来显示。CSS 语言是一个用于网页排版的标记性语言。

10.2.2 CSS 的发展与特点

1994 年哈坤·利提出了 CSS 的最初建议。伯特·波斯(Bert Bos)当时正在设计一个叫 Argo 的浏览器，他们决定一起合作设计 CSS。

当时已经有一些样式表语言的建议，但 CSS 是第一个含有“层叠”的思想的。在 CSS 中，一个文件的样式可以从其他的样式表中继承下来。设计者在有些地方可以使用他自己

更喜欢的样式，在其他地方则继承，或“层叠”作者的样式，这种层叠的方式使作者和读者都可以灵活地加入自己的设计，混合各人的爱好。

哈坤于1994年在芝加哥的一次会议上第一次展示了CSS的建议，1995年他与波斯一起再次展示这个建议。当时W3C刚刚建立，W3C对CSS的发展很感兴趣，为此组织了一次讨论会。哈坤、波斯和其他一些人(例如微软公司的托马斯·雷尔登)是这个项目的主要技术负责人。1996年底，CSS已经完成。1996年12月，CSS要求的第一个版本被出版。

1997年初，W3C内组织了专门管理CSS的工作组，其负责人是克里斯·里雷。这个工作组开始讨论第一版中没有涉及的问题，其结果是在1998年5月出版的第二版要求。到目前为止，第三版已经完备。

采用CSS+DIV进行网页重构相对于传统的TABLE网页布局而言具有以下3个显著优势：

1. 表现和内容相分离

将设计部分剥离出来放在一个独立样式文件中，HTML文件中只存放文本信息，这样的页面对搜索引擎更加友好。

2. 提高页面浏览速度

对于同一个页面视觉效果，采用CSS+DIV重构的页面容量要比TABLE编码的页面文件容量小得多，前者一般只有后者的1/2大小，浏览器就不用去编译大量冗长的标签了。

3. 易于维护和改版

用户只要简单地修改几个CSS文件就可以重新设计整个网站的页面。

10.2.3 CSS在网页中的应用

CSS最重要的作用是将HTML页面的内容和它的显示分隔开来。在CSS出现以前，几乎所有的HTML文件内都包含文件显示的信息，如字体的颜色、背景应该是怎样的，如何排列，边缘连线等必须一一在HTML文件中列出，有时甚至重复列出。CSS使HTML页面开发者可以将这些信息中的大部分隔离出来，简化了HTML文件，这些信息被放在一个辅助的用CSS语言写的文件中，HTML文件中只包含结构和内容的信息，CSS文件中只包含样式的信息。

CSS样式信息可以包含在一个附件中，也可以包含在HTML文件中，不同的媒体可以使用不同的样式表，例如一个文件在显示器上的显示可以与在打印机中打印出的显示不同，这样页面浏览者就有更大的控制显示的自由。

10.3 Dreamweaver CS6的应用

编写HTML和CSS有两种方法，一种是手工直接编写，即利用任何的文本编辑器打开并编写；另一种是利用可视化软件编写，如Frontpage、Dreamweaver等。

10.3.1 窗口界面

Adobe Dreamweaver CS6 是美国 Macromedia 公司开发的可视化网页编辑器，它将网页制作和网站管理集于一身，是目前常见的所见即所得的网页编辑软件。图 10-1 所示的是 Dreamweaver CS6 的窗口主界面。

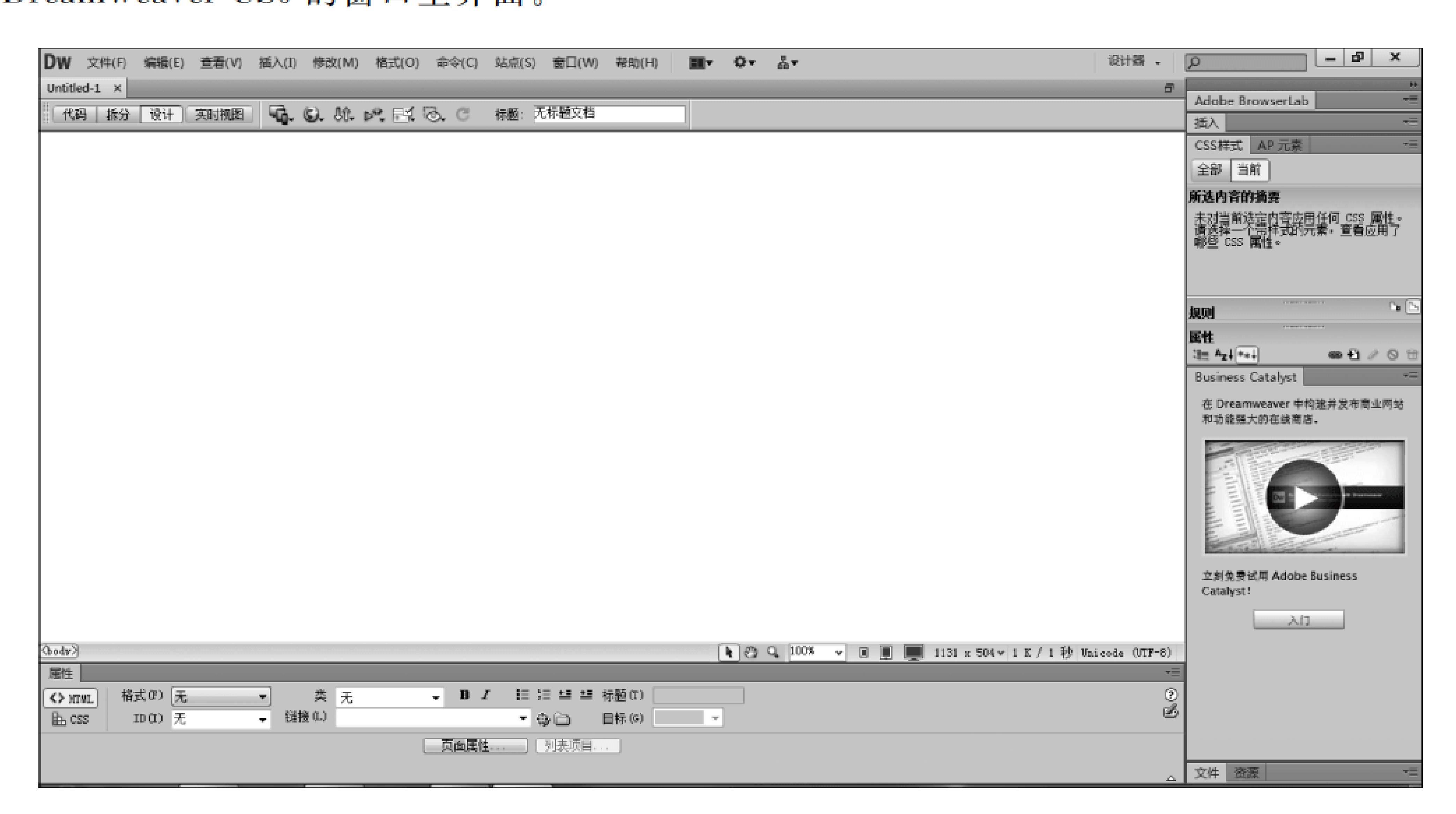


图 10-1 Dreamweaver CS6 窗口界面

该窗口由菜单栏、插入栏、文档工具栏、文档窗口、面板组及【属性】面板组成，具体如下：

- 菜单栏：Dreamweaver 中共包含文件、编辑等 10 个菜单。
- 插入栏：包含用于创建和插入对象的按钮。
- 文档工具栏：包含【代码】、【拆分】、【设计】等按钮。
- 文档窗口：用于显示当前文档。
- 面板组：列出 Dreamweaver 中的所有控制面板，可显示和隐藏。
- 【属性】面板：列出与当前选择内容相关的属性。

10.3.2 基本网页的制作

1. 创建站点

先建立一个文件夹，称为站点根目录，然后选择菜单栏中的【站点】|【新建站点】命令，在站点设置对话框中进行相应设置，如图 10-2 所示。

至此站点建立完毕，以后可以随时编辑修改这个站点的相关属性。在【文件】面板中右击已经建立的站点，然后新建文件夹，分别建立 index 和 image 文件夹，之后就可以把制作页面时用到的图、Banner、背景图等先放在 image 文件夹里，方便在需要插入图片等元素时使用。

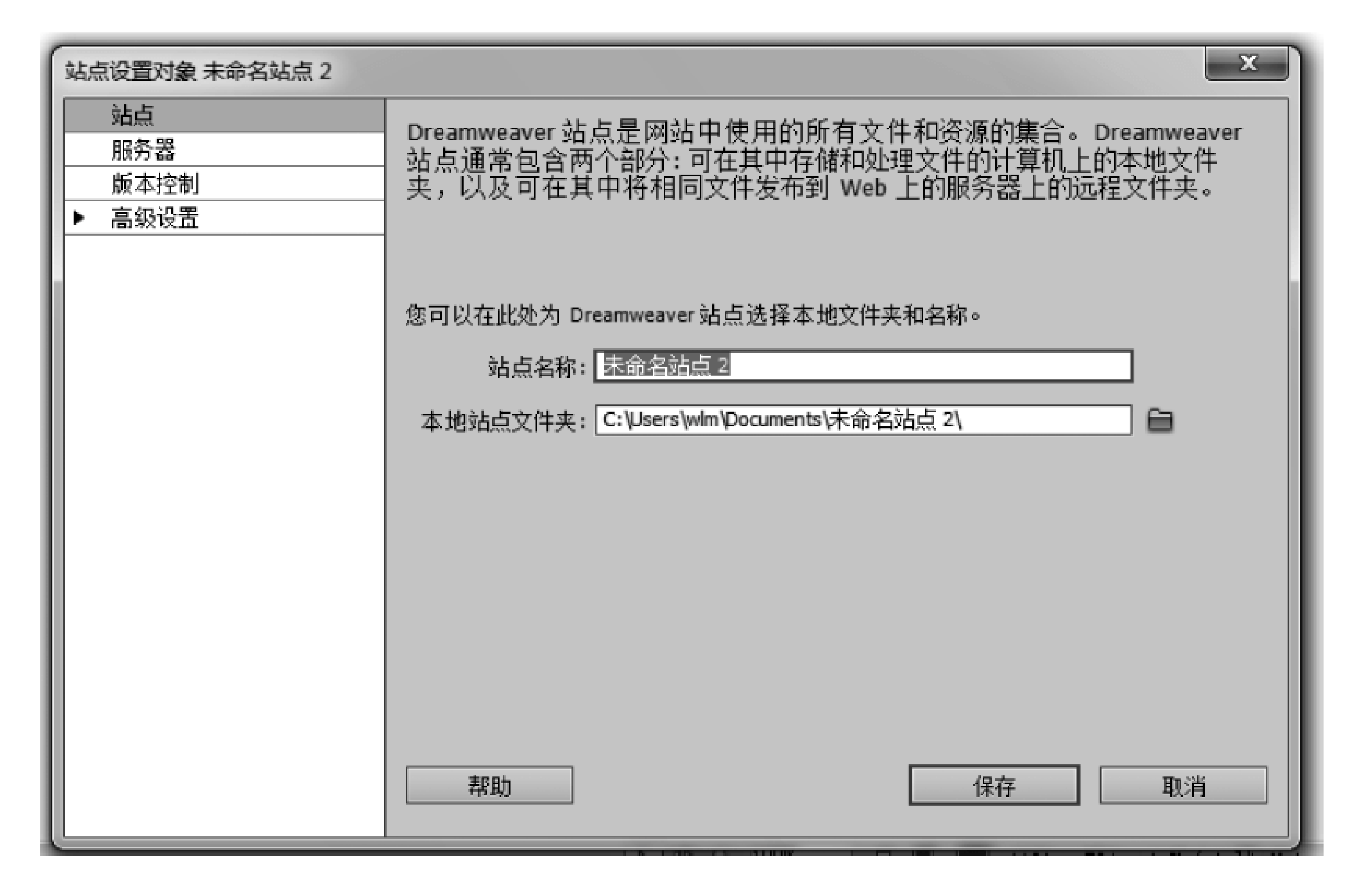


图 10-2　站点设置对话框

2. 新建网页

选择菜单栏中的【文件】|【新建】|【新建文档】命令，打开【新建文档】对话框，在其中选择新建的文档类型，如图 10-3 所示，单击【创建】按钮建立所需的文件。



图 10-3　【新建文档】对话框

3．设置网页属性

网页页面属性包括网页的“外观”、“链接”、“标题”、“标题/编码”和“跟踪图像”等。

选择菜单栏中的【修改】|【页面属性】命令，在弹出的【页面属性】对话框中设置页面属性，如图 10-4 所示。



图 10-4 【页面属性】对话框

- 【外观(CSS)】列表框：在其中对网页的字体、字号、字的颜色、背景图像及颜色、边距进行设置。
- 【外观(HTML)】列表框：设置背景图像及颜色、超链接的文本颜色、边距等。
- 【链接(CSS)】：设置链接的字体、字号、不同状态链接的颜色，以及下划线。
- 【标题(CSS)】：设置网页中标题的样式，即从标题 1 到标题 6 的样式。
- 【标题/编码】：设置本网页的网页标题、编码格式及文档类型等。
- 【跟踪图像】：设置跟踪图像的属性。跟踪图像一般在设计网页时作为网页背景，用于引导网页的设计。

4．文字

在网页中文字是最基本的组成元素，在输入文字后可以对文字进行相应的格式化操作，图 10-5 所示为文字的【属性】面板，在其中可以为文字设置格式、ID、类、链接、加粗、斜体、项目列表、编号列表、删除内缩区块、内缩区块、标题、目标、页面属性、列表项目等属性。

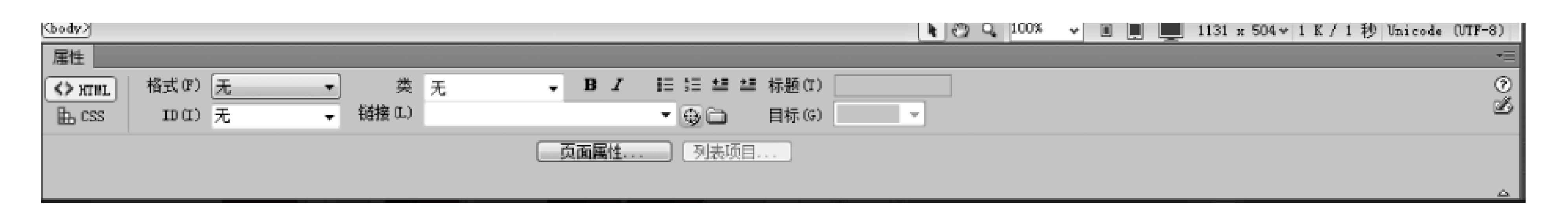


图 10-5 文字【属性】面板

5. 图片

Dreamweaver 中常用的图片格式有 3 种，即 GIF、JPEG、PNG。

1）插入图像

选择菜单栏中的【插入】|【图像】命令，在打开的【选择图像源文件】对话框中选择图像，单击【确定】按钮，如图 10-6 所示。

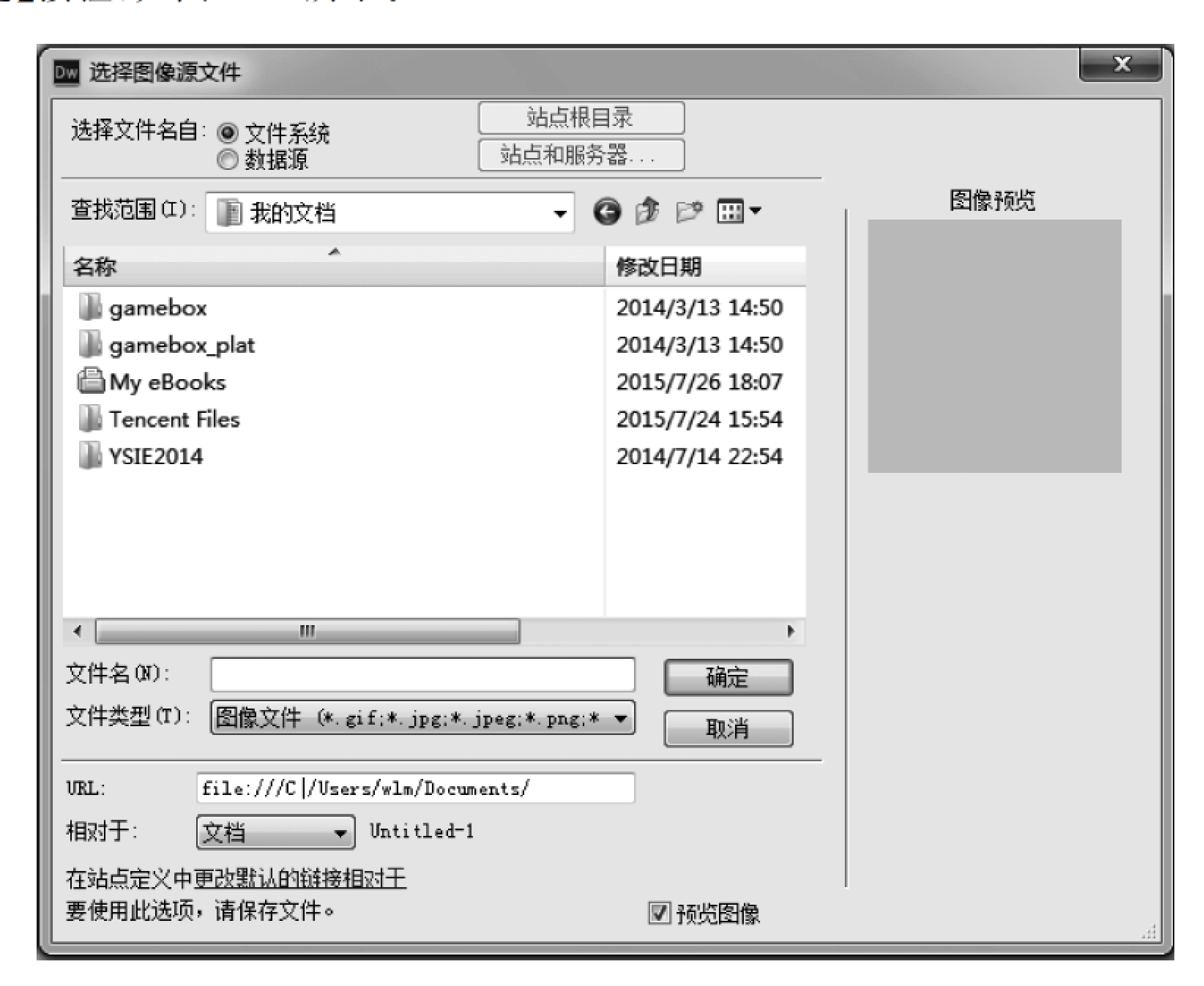


图 10-6　【选择图像源文件】对话框

2）图像【属性】面板

单击需要修改属性的图像，【属性】面板随之变为图像【属性】面板，如图 10-7 所示。

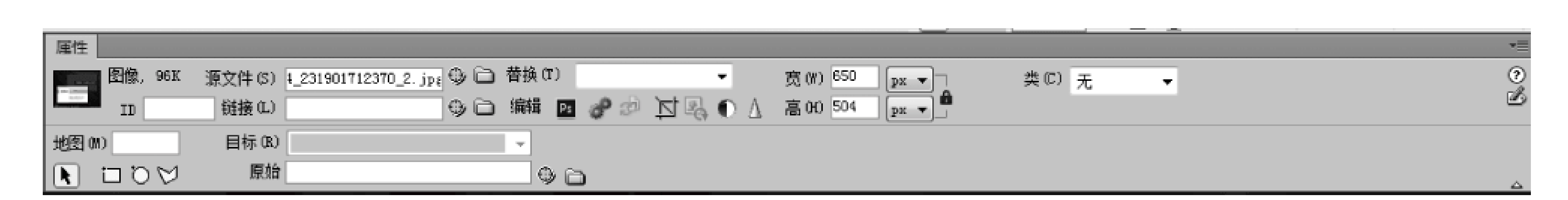


图 10-7　图像【属性】面板

其中包括图像 ID、宽、高、源文件、链接、替换、类、编辑、地图、垂直边距、水平边距、目标、原始、边框、对齐等基本设置。

3）图像热区链接

图像热区指在一幅图片上创建多个区域（热点），并可以单击触发。当用户单击某个热点时会发生某种链接或行为。

选中图像，在图像【属性】面板中使用热区工具（矩形、椭圆、多边形）在图像上划分热区，并为绘制的每一个热区设置不同的链接地址和替代文字。

6. 超链接

1）什么是超链接

所谓的超链接是指从一个网页指向一个目标的连接关系，这个目标可以是另一个网页，也可以是相同网页上的不同位置，还可以是一个图片、一个电子邮件地址、一个文件，甚至是一个应用程序。

2）创建超链接

（1）到网站内页面的超链接——内部链接（在网站内部页面之间创建相互链接关系）：

选中页面中的文字或图像，在【属性】面板中单击【链接】栏右侧的文件夹图标，通过浏览选择一个文件。

（2）到网站外页面的超链接——外部链接：

选中文字或图像之后，直接在【属性】面板的【链接】栏中输入外部的链接地址，例如“http://www.163.com”。

（3）到网页某一特定位置的超链接——锚点链接：

首先创建并命名锚点，就是在网页中设置位置标记，并给该位置一个名称，以便引用，然后在【属性】面板的【链接】栏中直接输入“#锚点名”。

（4）创建E-mail电子邮件链接：

选中文本或图像，在插入栏的【电子邮件链接】中输入邮件地址，或在【属性】面板的【链接】栏中直接输入“mailto:邮件地址”。

（5）创建空链接：

选中要制作空链接的对象，在【链接】栏中直接输入“#”。

3）链接的目标框架

从【目标】下拉菜单中选择打开文档的位置。

- _self：在当前网页所在的窗口或框架中打开（默认方式）。
- _blank：每个链接会创建一个新的窗口。
- _new：在同一个刚创建的窗口中打开。
- _parent：如果是嵌套的框架，则在父框架中打开。
- _top：在完整的浏览器窗口中打开。

10.3.3 表格

1. 表格的作用

（1）存放数据。

（2）布局页面。

2. 插入表格

步骤如下：

（1）单击网页中需要插入表格的地方。

（2）在菜单栏中选择【插入】|【表格】命令，或者单击【常用】工具栏里的【表格】按钮，或

者按 Ctrl+Alt+T 键，打开图 10-8 所示的对话框，在对话框中对表格大小、行数和列数、表格宽度、单元格边距、单元格间距进行设置。



图 10-8　【表格】对话框

3. 表格【属性】面板

选中表格，在表格四周会出现 3 个控制点，此时【属性】面板为表格【属性】面板，如图 10-9 所示。

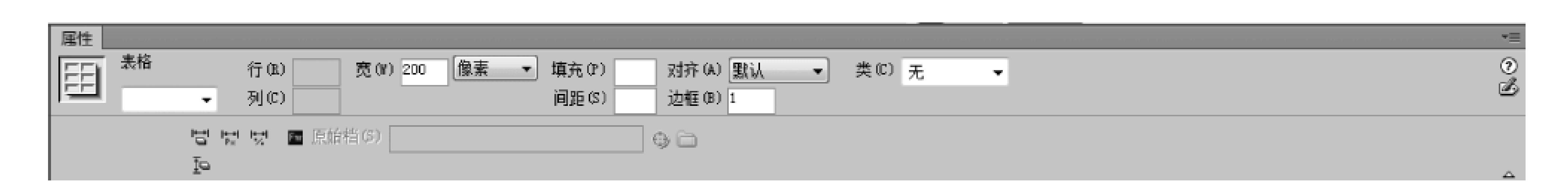


图 10-9　表格【属性】面板

其中包含有表格 ID、行数、列数、宽、填充、间距、类、边框、对齐、清除列宽、清除行高、将表格宽度转为像素、将表格宽度转为百分比等参数。

10.3.4　框架

1. 建立框架

选择菜单栏中的【修改】|【框架集】命令，在其级联菜单中选择一种框架样式，分别对每个框架内的网页进行编辑。

2. 框架的属性

首先单击【窗口】菜单，选择【框架】命令，打开【框架】面板，然后在【框架】面板上选择框架，选择框架后可以在框架【属性】面板上查看框架的属性，如图 10-10 所示。

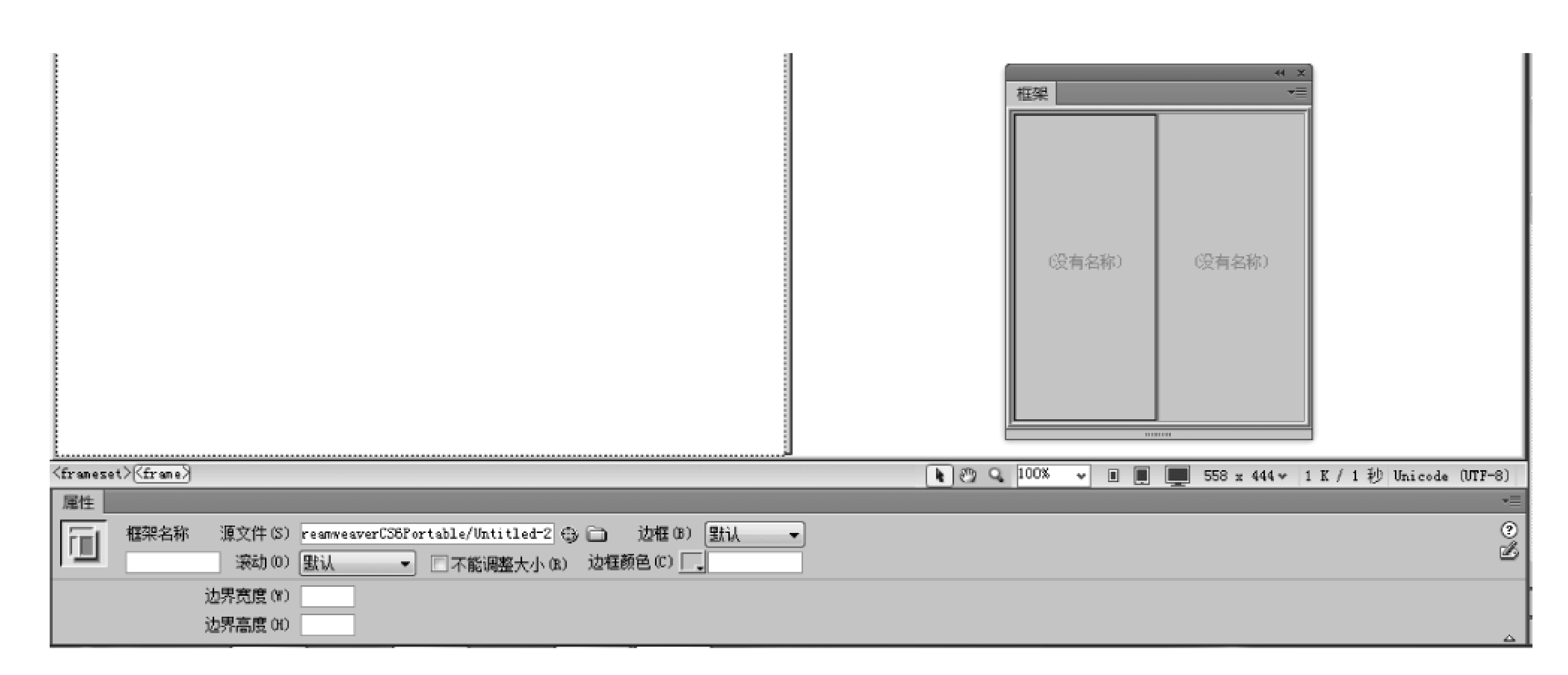


图 10-10 框架【属性】面板

其中包括框架的源文件、框架名称、框架边框、框架是否滚动、框架边距、边框颜色以及【不能调整大小】复选框。

3. 框架集的属性

首先单击【窗口】菜单，选择【框架】命令，打开【框架】面板，然后在【框架】面板上选择框架集，如图 10-11 所示，或在网页的框架上单击，也可以将【属性】面板显示为框架集的【属性】面板。

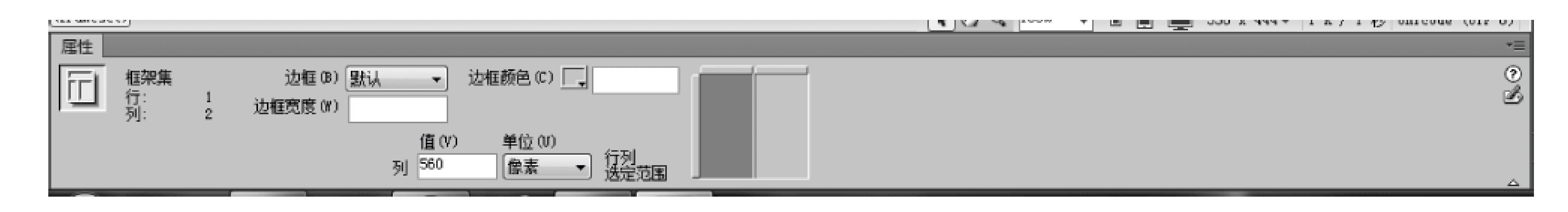


图 10-11 框架集【属性】面板

其中包括边框宽度、框架大小、边框颜色以及是否显示“边框”。

4. 保存框架和框架集

1）保存框架

将鼠标指针置于要保存的框架，选择菜单栏中的【文件】|【保存框架】命令。

2）保存框架集

选择要保存的框架集，然后选择菜单栏中的【文件】|【保存框架页】或者【文件】|【框架集另存为】命令。

3）保存全部

这是保存整个框架结构，在保存的时候虚线罩住的部分就是用户现在保存的框架。

5. 框架删除

拖动不需要的边框和其他边框合并，或者拖到编辑窗口外。

10.3.5　表单

1. 什么是表单

表单是用户与计算机交互的一种有效手段，其中包含文本字段、密码字段、单选按钮、复选框等表单对象。当浏览者在浏览器的表单内填写信息并单击【提交】按钮后，这些信息会被发送至服务器，服务器中的服务器端脚本或应用程序对这些信息进行处理，以此进行响应。

2. 创建表单域

1）插入表单

选择菜单栏中的【插入】|【表单】|【表单】命令，出现红色虚线框，此时该表单内的所有内容都将出现在此红色虚线框内。

2）表单【属性】面板

插入表单后，【属性】面板改为表单【属性】面板，如图 10-12 所示。



图 10-12　表单【属性】面板

该面板中包括了表单 ID、动作、方法、目标、编码类型、类等参数。

- 【动作】：表单处理的程序。
- 【方法】：表单提交数据的方法，共 POST 和 GET 两种。

3. 创建表单对象

选择菜单栏中的【插入】|【表单】命令，在级联菜单中选择需要的表单对象。表单对象包括文本框、密码框、单选按钮、复选框、提交按钮、重置按钮、图像按钮、文件域、隐藏域、下拉列表、文本域等。

10.3.6　声音和动画

1. 声音

经常使用的声音格式有 MP3 格式、RealAudio 格式、WMA 格式、MID 格式。

1）背景音乐

直接插入 HTML 标记：

```
<bgsound src="11.mp3" autostart=true loop=-1>
```

2）将插件嵌入到网页中

选择菜单栏中的【插入】|【媒体】|【插件】命令，在【属性】面板中单击【链接栏】文本框旁边的文件夹图标以浏览音频文件，或者在【链接栏】文本框中输入文件的路径，通过在适当的文本框中输入值或者通过在文档窗口中调整插件占位符的大小，输入宽度和高度，这些值确定音频控件在浏览器中以多大的尺寸显示。

2. 动画

在网页中可以使用动画，包括 GIF 动画和 Flash 动画，其中 Flash 动画的应用相当广泛，要想在网页中应用 Flash 动画，需要将 Flash 动画导出为 SWF 格式才能插入到网页中。

选择菜单栏中的【插入】|【媒体】|SWF 命令，打开【选择 SWF】对话框，在其中找到需要的动画，单击【确定】按钮即可。

10.3.7 CSS

1. 创建样式

选择菜单栏中的【窗口】|【CSS 样式】命令，调出 CSS 样式的浮动面板，如图 10-13 所示；单击面板下方的【新建 CSS 样式】按钮，在弹出的【新建 CSS 规则】对话框（如图 10-14 所示）中选择样式类型、命名，单击【确定】按钮；在【CSS 规则定义】对话框中选择具体样式内容，如图 10-15 所示。

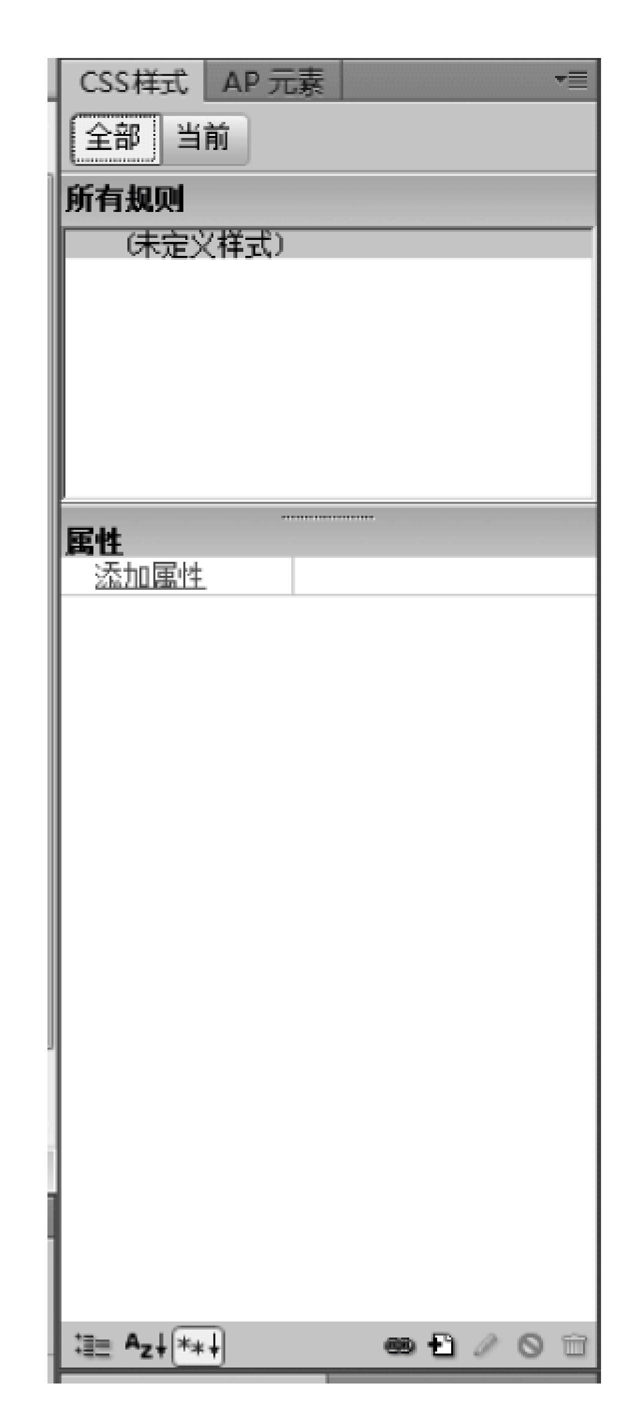


图 10-13　CSS 样式的浮动面板



图 10-14　【新建 CSS 规则】对话框

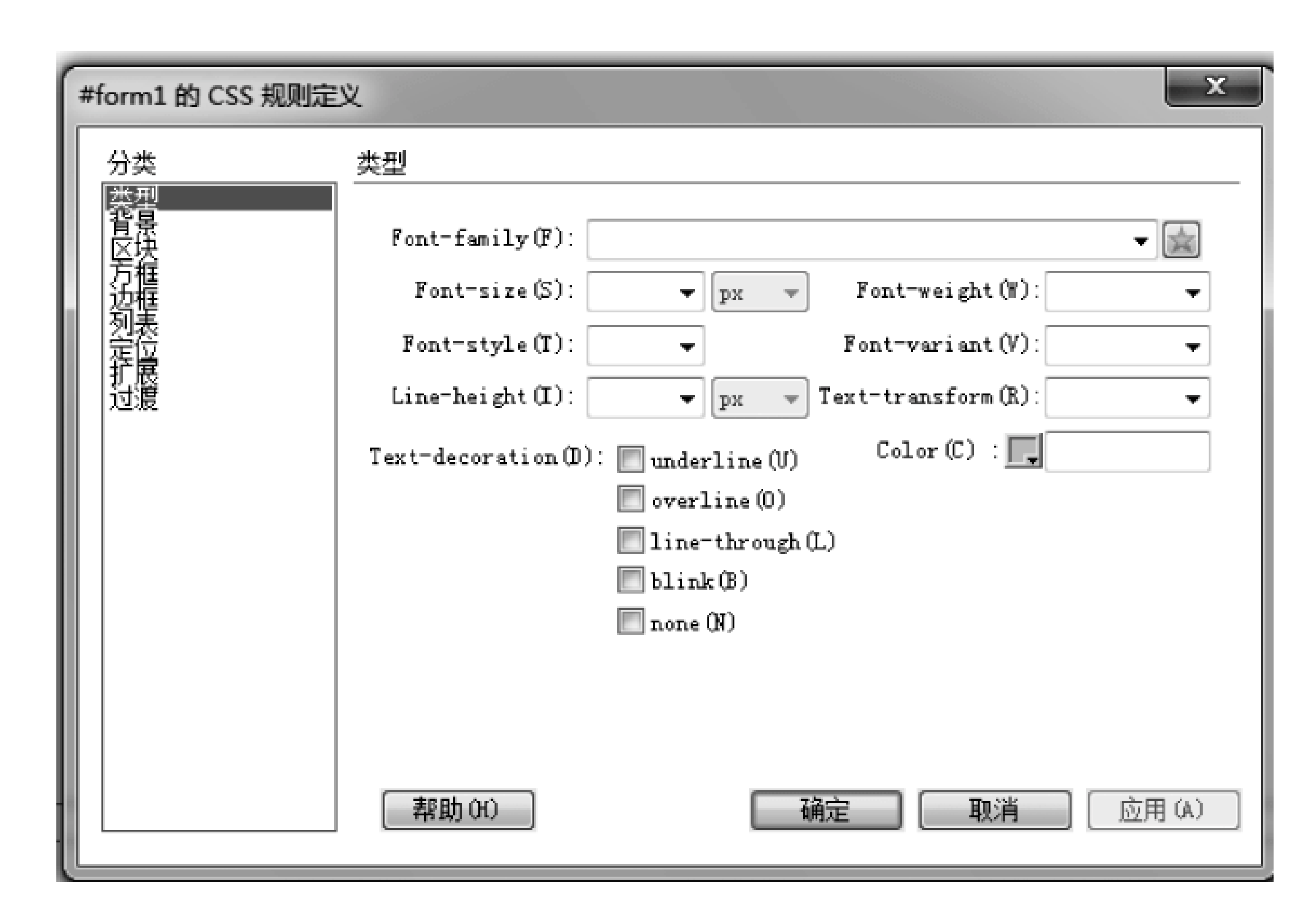


图 10-15　【CSS 规则定义】对话框

2. 应用样式

如果是外部样式文件,需要链接。如果是内部样式文件,需要选中相应文字,套用。

第11章 HTML

本章学习目标：

✍了解标记及规则。

✍掌握 HTML 中的常用标记及属性。

✍掌握 HTML 网页的制作。

11.1 HTML 基础

一个 HTML 文件就是一个静态网页文件，既可以用记事本编辑，也可以用 Dreamweaver 编辑。

11.1.1 一个简单的 HTML 实例

下面用记事本编写第一个 HTML 文件，步骤如下：

(1) 单击【开始】按钮，选择【程序】|【附件】|【记事本】命令，打开记事本程序。

(2) 在记事本中直接输入下面的内容。

```
<html>
<head>
<title>这是第一个 HTML 页</title>
</head>
<body>
hello world!!!!
</body>
</html>
```

(3) 选择菜单栏中的【文件】|【保存】命令，在打开的【另存为】对话框中为文件选择保存位置，输入文件名，将文件类型手工改为.html，如图 11-1 所示，单击【保存】按钮。

(4) 用 IE 浏览。

11.1.2 HTML 的基本结构

HTML 网页文件包含两个部分，即文件头和文件体。在文件头里，主要是对这个 HTML 文件进行一些必要的定义，文件体中的内容才是真正要显示的各种文件信息。在一

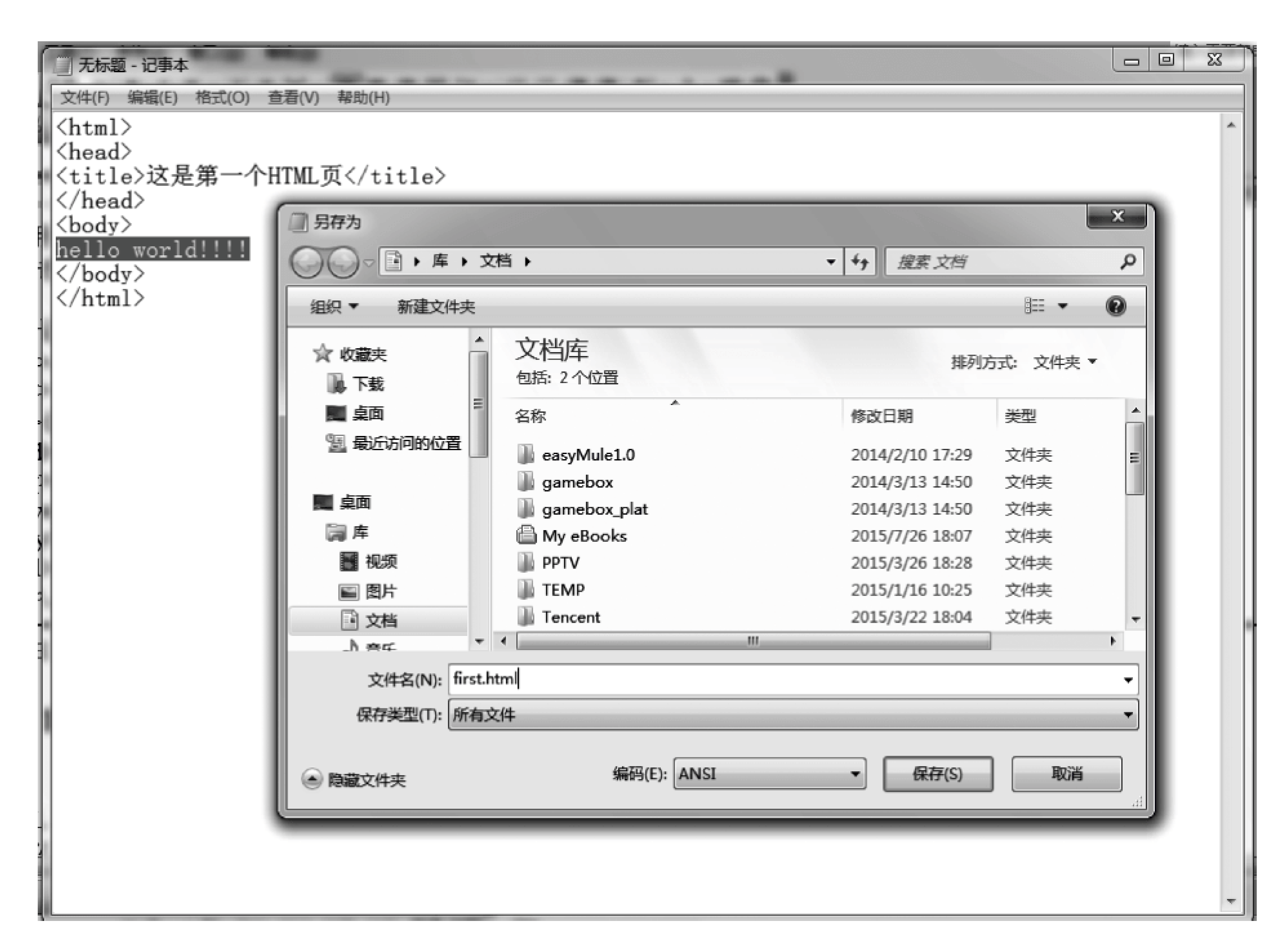


图 11-1 用记事本制作网页

个 HTML 文件中包含各种 HTML 元素，如图片、表格、表单等，这些元素在页面中使用标记分隔，可以说 HTML 文件是由各种 HTML 元素和标记组成的。下面是 HTML 基本结构：

```
<html>
<head>
<title></title>
</head>
<body>
</body>
</html>
```

在 HTML 文档中，第一个标签是<html>，这个标签告诉浏览器这是 HTML 文档的开始。HTML 文档的最后一个标签是</html>，这个标签告诉浏览器这是 HTML 文档的终止。在<head>和</head>标签之间的文本是头信息，在浏览器窗口中，头信息是不被显示的。<title>和</title>标签之间的文本是文档标题，它们被显示在浏览器窗口的标题栏中。在<body>和</body>标签之间的文本是正文，会被显示在浏览器中。

11.1.3 HTML 的基本标记

1. 文件类型标记

文件类型标记是<html>，是双标记，用来标记该文件是 HTML 类型的文件，位于 HTML 文件的最外层，也就是说这一对标记必须放置在代码的开头和结尾，其语法如下：

```
<html>
</html>
```

HTML 标记是最基本的标记，它标志着这段代码是用 HTML 语言来描述的。

2. HTML 头标记

HTML 的头标记以<head>开始、以</head>结束，是一个双标记，其语法如下：

```
<head>头部内容</head>
```

它用于包含当前文件的相关信息，一般包括标题、基底信息、元信息等。在一般情况下，CSS 样式也是定义在头元素中。定义在头部的内容一般不会在网页中直接显示，而是通过另外的方式起作用。表 11-1 所示为在头部常用的标记。

表 11-1　头部的常用标记

标　记	描　述
<base>	当前文档的 url 全称(基底网址)
<basefont>	设定基准的文字字体、字号和颜色
<title>	设定显示在浏览器左上方的标题内容
<isindex>	表明该文档是一个可用于检索的网关脚本，由服务器自动建立
<meta>	有关文档本身的原信息，例如用于查询的关键词、用于获取该文档的有效期等
<style>	设定 CSS 层叠样式表的内容
<link>	设定外部文件的链接
<script>	设定页面脚本程序的内容

3. HTML 主体标记

文件类型标记是<body>，是双标记，用于包含当前文件的页面内容，也就是说在该标记之间的内容是页面中真正要显示的内容，包括文字、图片、表格等。

在<body>标记中可以包含多种属性，用于设置页面的背景、字体等属性。其语法如下：

```
<body bgcolor="颜色" background="图片名" bgproperties="fixed" text="颜色" link="颜色"
vlink="颜色" alink="链接按钮" topmargin=距离 leftmargin=距离>
网页内容
</body>
```

说明：

- bgcolor：设置背景颜色。
- background：设置背景图片。
- bgproperties：设置背景图片固定不变。
- text：设置文本颜色。
- link：默认链接。
- vlink：当鼠标按下时的链接。
- alink：当鼠标松开后的链接。

- topmargin：文本距顶部的距离。
- leftmargin：文本距左边的距离。

4. 页面标题

页面标题标记是一个特殊的标记，它设置的内容并不显示在页面中，而是显示在浏览器的标题栏中，用来说明文件的用途。因此，在设置此标记的时候，要使其能够体现整个页面的主题。

在 HTML 文件中，标题信息设置在页面的头部分，也就是位于＜head＞＜/head＞标记之间。标题以＜title＞开始、以＜/title＞结束，是一个双标记，其语法如下：

```
<title> 标题内容</title>
```

说明：在 HTTML 文件中，页面的标题只能有一个，用于帮助浏览者更好地识别页面。

11.2 文字、列表与图片

11.2.1 文字

网页中的文字是一个网页最基础的部分，而文字格式主要指用一些 HTML 元素来标记文本的方式，用来更改文字的大小、颜色、字体、加粗、斜体等。

1. 文字的基础格式

在 HTML 中，＜font＞元素可以用来显示文字的属性，包括文字的颜色、大小和字体等。其具体语法如下：

```
<font color = "颜色" face = "字体名称" size = "n">文字内容</font>
```

说明：

- color：定义文字的颜色，可以是颜色的英文名称，也可以是十六进制的颜色代码。
- face：定义文字的字体。
- size：定义文字的大小，有效范围是 1～7。

2. 加粗和斜体

通常，在进行文字处理时都会对比较重要的内容使用加粗、斜体来引起读者注意，其具体语法如下：

```
<b>加粗的文字</b>
<i>倾斜的文字</i>
<strong>加粗的文字</strong>
<em>倾斜的文字</em>
```

说明：从运行结果看，＜b＞与＜strong＞对于加粗的文字效果没有什么区别，而＜i＞与＜em＞使文字倾斜的效果也没有什么区别。

3. 下划线与删除线

有些时候网页设计者希望文字等带有下划线表示突出显示，或者在修改文字的上面使用删除线表示文字被删除，其具体语法如下：

```
<u>下划线</u>
<strike>删除线</strike>
<del>删除线</del>
<ins>插入线</ins>
```

说明：其中<u>与<strike>元素在 HTML4.0 以上版本中不建议使用。

4. 上标与下标

在描述一些复杂的表达式时，特别是一些数学公式时，经常会用到上标与下标，其具体语法如下：

```
<sup>上标文字</sup>
<sub>下标文字</sub>
```

5. 等宽文字

等宽字体一般针对英文字体而言，其具体语法如下：

```
<tt>英文文字</tt>
```

6. 闪烁文字

这是除了粗体和斜体之外的另一种使文字突出显示的方式，其具体语法如下：

```
<blink>闪烁文字</blink>
```

说明：需要注意的是该元素在 IE 浏览器中不支持，其他如 Netscape、Opera、Firefox 等浏览器支持。

11.2.2 段落

1. 段落

在 HTML 中，使用<p>元素区分一个与另一个段落，在<p>与</p>标签内的文字就是一段，其具体语法如下：

```
<p>一段文字</p>
```

说明：其中</p>可以省略。

2. 段落居中

要想居中显示网页中的段落，可以使用<center>元素，其具体语法如下：

```
<center>设置元素的居中对齐</center>
```

说明：其中的元素包括文字、图像、表格等。

3. 预定义格式

在HTML源代码中，即使文字已经换行，但是只要没有使用
标签，在浏览器中显示出来的文字就不会被换行；如果想要在浏览器中显示出源代码里所设置的所有格式，如空格、制表符等，可以使用<pre>元素，<pre>元素相当于设置了一个“块”，这个块可以将源代码里的所有文本（除HTML标签外）在浏览器中按原样显示出来。其具体语法如下：

```
<pre>设定了格式的文字</pre>
```

说明：该标记是将其中的所有文本原样显示，包括文字间的空白，如空格、制表符等。

4. 文本缩进

<blockquote>主要用于设置文本的缩进效果，通过<blockquote>元素可以实现页面文字的缩进，从而使页面的文字布局更加错落有致。其具体语法如下：

```
<blockquote>需要进行缩进的文字</blockquote>
```

说明：本标记可以嵌套使用。

5. 换行

是在HTML中人为换行的元素，一次只能换一行，其具体语法如下：

```
<br>
```

说明：该标记为单向标记，标记前后的内容不在一行。

6. 水平分隔线

当页面内容比较烦琐时，可以在段与段之间插入一条水平线来使页面显得层次分明，便于阅读。插入水平线的具体语法如下：

```
<hr align="对齐方式" width="宽度" size="高度" color="颜色" noshade>
```

说明：该标记为单向标记，在标记位置行成一水平线。

11.2.3 列表

1. 无序列表

无序列表是一个对列表项目出现次序不做要求的列表，列表项目之间是并列关系，不存在先后次序。当浏览器显示无序列表时会在列表项目前加上一个项目符号，而不是显示数字，该符号也可以由网页开发人员指定。

其具体语法如下：

```
<ul type="符号取值">
<li>无序列表项 1</li>
…
<li>无序列表项 n</li>
</ul>
```

说明：type 属性可以取 3 个值，即“disc”、“square”、“circle”(在某些浏览器中，type 属性的值必须使用小写浏览器才可以识别)，用来控制强调符风格的属性，3 个值分别对应实心圆点、实心方块、空心圆圈。需要说明的是有些浏览器不支持该属性。

2. 有序列表

与无序列表相对应的是有序列表，有序列表中的列表项目通常是有先后次序的，并且不能随意更换这些次序。当浏览器显示有序列表时会在列表项目前加上一个编号，用来标识项目出现的次序，当然该编号也可以由网页开发人员指定。

其具体语法如下：

```
<ol type ="符号取值" start ="">
<li>有序列表项 1</li>
…
<li>有序列表项 n</li>
</ol>
```

说明：

- type 属性可以取 5 个值，即“1”、“a”、“A”、“i”、“I”，用来控制强调符风格的属性，5 个值分别对应数字、小写字母、大写字母、小写罗马数字、大写罗马数字。
- start 属性是一个数字，表示从第几个数字(字母)开始计数。

3. 定义列表

定义列表也称为字典列表，它是一种包含两个层次的列表，主要用于进行名词解释或名词定义，名词是第一层次，其解释或定义是第二层次。另外，这种列表不包括项目符号，每个列表项带一段缩进的定义文字。其语法如下：

```
<dl>
<dt>名词 1</dt>
<dd>名词解释 1</dd>
<dt>名词 2</dt>
<dd>名词解释 2</dd>
…
</dl>
```

说明：<dl>标记表示一个定义列表的开始；</dl>表示定义列表的结束；<dt>表示要被解释的名词；<dd>表示这段文字是对前面名词的解释说明。

11.2.4 图片

在一个网页中可以插入 Logo(网站标志)、Banner(横幅广告)等各种图片，当浏览者浏

览网页时，这些图片会自动显示出来。合理地应用图片可以使网页看起来更生动、形式更活泼。在 HTML 中可以通过<img>元素插入图片，其语法代码如下：

```
<img src="url" alt="替代文字" name="名字" width="宽度" height="高度" border="边框"
aligh="对齐方式" id="编号">
```

说明：

- src：用来指定图片的位置，可以是相对路径也可以是绝对路径。
- alt：指定用于替代图片的文本，当图片不能正常显示时，可以使用替代文本来替代图片。
- name：图片的名称，在很多时候可以省略。
- width：指定图片的宽度。
- height：指定图片的高度。
- border：指定图片的边框大小，该属性的值越大，边框越粗。
- aligh：用于设置图片的对齐方式，它有 5 个属性，即 left、righ、top、middle 和 bottom，分别表示左对齐、右对齐、顶部对齐、中间对齐和底部对齐。
- id：图片的编号，也可以省略，在同一个 HTML 文档中不允许出现相同的 id 号，但可以出现相同的 name。

11.2.5 网页背景

1. 网页背景颜色

用户可以在<body>元素中设置背景颜色，其语法如下：

```
<body bgcolor="背景颜色">网页内容</body>
```

说明：背景颜色和文字颜色的表示方法一样，可以是颜色的英文名称，也可以是十六进制的颜色代码。

2. 网页背景图片

网页背景除了可以用颜色之外，还可以用图像，同样是在<body>元素中设置。其语法如下：

```
<body background="背景图片地址">网页内容</body>
```

说明：背景图片可以是本地文件夹内的，也可以是网络中的图片，如果是网络中的图片，可以用 URL 地址指定图片位置。

11.3 超链接

超链接是网站中最重要的组成部分，HTML 有了超链接才显得与众不同。超链接可以方便用户的使用，允许浏览者从一个网页跳转到另一个网页，多个网页因为有了超链接才形成一个网站。超链接不仅可以链接网页，还可以链接图片、视频、音频，甚至是任何

一种文件。

11.3.1 创建超链接

在实际应用中很少有网页是单独的,通常会使用超链接创建一个页面与其他页面之间的联系,同样也可以使用超链接创建与其他 Web 服务器上的网页联系。使用超链接不仅可以链接网页文件,还可以链接其他文件。

1. 超链接标记

在 HTML 中,创建超链接的标记是<a>。<a>标记是双标记,以<a>开始,以结束。<a>标记创建的链接能指向一个 HTML 页面、一幅图像等任何资源,其语法如下:

```
<a name = "锚名称" href = "URL" title = "标题" target = "目标">超链接文字</a>
```

说明:超链接元素的属性有很多,含义如下。

- name:用于设置超链接当前位置的锚名称。
- href:用于设置超链接的链接地址。
- title:用于设置超链接的标题。
- target:用于设置打开超链接的目标地址。

2. 链接地址

链接地址用于设置超链接的路径,可以使用<a>标记中的 href 属性进行设置。设置链接地址的语法如下:

```
<a href  = "链接地址">用来设置超链接的元素</a>
```

说明:在这段语法中,超链接的链接地址可以是相对地址,也可以是绝对地址。

绝对路径实际上就是完整路径。这种路径可以是按照硬盘文件的真正路径,也可以是按照域名的完整网页路径。使用绝对路径定位链接目标文件比较清晰,但是如果该文件被移动了,就需要重新设置所有的相关链接。

相对路径顾名思义就是自己相对于目标位置的路径,在使用相对路径的时候,不论将这些文件放到哪里,只要它们的相对关系没有变,就不会出错。一般有以下 3 种相对路径的写法。

- 同一目录下的文件:只需要输入链接文件的名称即可,例如“01.html”。
- 上一级目录中的文件:在目录名和文件夹名之前加入“../”,例如“../05/01.html”。
- 下一级目录:输入目录名和文件名,之间以“/”隔开,例如“html/05/01.html”。

3. 打开链接的方式

当单击一个网页里的超链接时,通常会在当前窗口中打开超链接目标页面,如果想保留当前网页的内容,让链接的页面从一个新建窗口里打开,应该怎么办呢?使用<a>元素的 target 属性可以实现这个功能。target 属性用来设置打开链接的方式,其语法如下:

```
<a target="目标">超链接文字</a>
```

说明：在 HTML 语言中，超链接的 target 属性可以取 4 个值，这些值的具体含义如表 11-2 所示。

表 11-2 target 属性值

属 性 值	表示的含义
_parent	在上一级窗口打开（常在框架页面中使用）
_blank	新建一个窗口打开
_self	在同一窗口打开，默认取值
_top	忽略所有的框架结构，在浏览器的整个窗口打开

11.3.2 锚点

在超链接中有一种特殊的链接形式，称为锚点链接。如果一个网页包含的内容很多，要想在整个网页里快速查找到自己感兴趣的内容，就不会那么方便了，这时可以通过锚点帮助用户方便地到达当前网页的其他位置。

1. 创建锚点

要使用锚点引导浏览者，首先要创建页面中的锚点。创建的锚点将确定链接的目标位置，其语法如下：

```
<a name="锚点名称">锚点的链接文字</a>
```

在这里，通过锚点名称可以标注相应的锚点，该属性是设置锚点必需的，锚点的链接文字有助于用户区分不同的锚点。在实际应用中可以不设置链接文字，这是因为设置的锚点仅仅是为链接提供一个位置，浏览页面时并不会在页面中出现锚点的标记。

2. 链接到本页的锚点

如果要链接到本页的命名锚点，只要在<a>元素的 href 属性中指定锚点名称，并在该名字前加上“#”字符。链接到本页锚点的语法如下：

```
<a href ="#锚点名称">锚点的链接文字</a>
```

3. 链接到其他网页的锚点

通常在单击一个超链接时都会打开一个网页，并且默认显示该网页的顶端，而不会是网页的底部或网页的其他位置。如果要打开一个网页，并且显示网页的某个区域，就必须创建锚点。使用<a>元素的 name 属性可以在网页上设置链接到其他网页的锚点，其语法如下：

```
<a href="页面地址#锚点名称">用于链接锚点的文字</a>
```

11.3.3 图像的超链接

<a>标记不仅可以为文字设置超链接，还可以为图片设置超链接。为图片设置超链接有两种方式，一种是将整个图片设置为超链接，只要单击该图片，就可以跳转到链接的URL上；另一种是为图片设置热点区域，将图片划分为多个区域，单击图片的不同位置将会跳转到不同的链接上。

1. 将整个图像设为超链接

将整个图像设为超链接的方法很简单，只需要将<a>标记中的文字换成<img>元素，在<img>元素中添加需要设置为超链接的图片即可。其语法如下：

```
<a href="链接地址"><img src="源文件地址"></a>
```

说明：其中两个地址都可以用相对路径也可以用绝对路径。

2. 设置图像热点区域

除了可以为整个图像设置超链接外，还可以为图像设置热点区域，也就是将一个图片划分成多个可单击的区域，单击不同的区域将跳到不同的链接上。在定义图像热点区域的时候，除了要定义图像热点区域名称之外，还要设置其热区范围，用户可以使用img元素中的usemap属性和<map>标签来创建，其语法如下：

```
<img src="url" usemap="#map名"><map name="map名"><area shap="图像热区形状" coords
="热区坐标" href="链接地址">
</map>
```

说明：usemap属性值中的“map名”必须是<map>标签中的name属性值，因为可以为不同的图片创建单击区域，每个图片都会对应一个<map>标签，不同的图片以usemap的属性值来区别不同的<map>标签。需要注意的是，usemap属性值中的“map名”前面必须加上“#”。

<map>标签里至少要包含一个<area>元素，如果一个图片上有多个可单击区域，将会有多个<area>元素。在<area>元素里，必须指定coords属性，该属性值是一组用逗号隔开的数字，通过这些数字可以决定可单击区域的位置，但是coords属性值的具体含义取决于shape的属性值，shape属性用于指定可单击区域的形状，其值如下。

- circle：指定可单击区域为圆形，此时coords的值应该是类似于“x,y,z”的表示方法，其中x和y代表圆心的坐标，该坐标是相对于图片的左上角而言的，也就是说图片左上角的坐标是“0,0”；而z代表圆的半径长度，单位为像素，circle也可以简写为circ。
- polygon：指定可单击区域为多边形，此时coords的值应该是类似于“x1,y1,x2,y2,x3,y3……”的表示方法，其中x1和y1是多边形的一个顶点的坐标，x2和y2是多边形的另一个顶点的坐标，至少有3个顶点才能形成一个区域(三角形区域)。同样这些坐标点也是相对于图片左上角而言的。在HTML里多边形是会自动闭合的，所以在coords里不用重复第一个坐标来闭合整个区域，polygon也可以简写成poly。

- rectangle：指定可单击区域为矩形，此时 coords 的值应该是类似于“x1，y1，x2，y2”的表示方法，其中 x1 和 y1 是矩形的一个角的顶点坐标，x2 和 y2 是该角的对角的顶点坐标，同样这些坐标点也是相对于图片左上角而言的，rectangle 也可以简写成 rect。

11.4 表格基础

在文档处理中，表格是一种很常用的表现手法。在 HTML 中表格除了用来对齐数据之外，更多地用来进行页面排版，无论是普通的 HTML 页面还是动态网站都需要使用表格来进行页面的布局。

11.4.1 创建表格

表格的开始元素是<table>、结束元素是</table>，所有的表格内容都位于这两个元素之间。如果要创建一个完整的表格，除了包含表格元素外，还要有行元素<tr>和单元格元素<td>。可以说，如果要在页面中创建一个完整的表格，至少要包含这 3 个元素。创建表格的具体语法如下：

```
<table>
<tr>
<td>表格的内容</td>
</tr>
</table>
```

说明：其中<tr>表示行，<td>表示列，上述代码表示在一行中有一列。

11.4.2 表格的属性

在默认情况下，表格只是作为布局的手段，因此不会在页面中显示出来，但有时候将表格显示出来并设置一定的效果能使页面内容更加协调。

1. 表格宽度

在默认情况下，表格的宽度是以内容为标准的，如果要设置表格的宽度为某一特定值而与其中的内容无关时，则可以使用 width 属性，其语法如下：

```
<table width="表格宽度">
<tr>
<td>表格的内容</td>
</tr>
</table>
```

说明：表格宽度可以是表格的绝对宽度，单位是像素；也可以设置为相对宽度，即相对于窗口的百分比。

2. 表格高度

除了可以为表格指定宽度之外，还可以为表格指定高度。通常，表格的高度都是由表格

的行数及单元格里的内容决定的，在为表格指明高度之后，如果表格的行数与单元格里的内容使表格的高度高于指定的高度，则浏览器将以实际的高度显示表格；如果实际高度低于指定高度，则浏览器以指定高度显示表格。<table>元素里的 height 属性值可以用来指定表格的高度，其语法如下：

```
<table height = "表格高度">
<tr>
<td>表格的内容</td>
</tr>
</table>
```

3. 表格背景图片

通过<table>元素中的 background 属性可以为表格指定一个背景图片，这种方法有点类似于为网页指定背景图片。如果背景图片小于表格大小，则会平铺该背景图片以充满整个表；如果背景图片大于表格大小，则会对背景图片进行裁剪，以适应该表格。其语法如下：

```
<table background = "图像源文件地址">
<tr>
<td>表格的内容</td>
</tr>
</table>
```

说明：background 的属性值也是一个标准的 URL 值，其图片类型可以是.gif、.png 或.jpeg 格式。

4. 单元格间距

单元格间距指表格中两个相邻单元格之间的间距和单元格与表格边框的间距。在默认情况下，单元格与单元格之间的间距是两个像素。通过设置<table>元素的 cellspacing 属性可以增大或减小单元格之间的间距。其语法如下：

```
<table cellspacing = "间距大小">
<tr>
<td>表格的内容</td>
</tr>
</table>
```

5. 表格内单元格与文字的距离

表格内单元格与文字的距离指在单元格内文字与单元格边框的距离，在默认情况下，文字是紧贴着单元格的边框出现的，这样会显得页面的内容有些拥挤，可以通过<table>元素中的 cellpadding 属性来调整这一距离。其语法如下：

```
<table cellpadding = "单元格与文字距离的值">
<tr>
<td>表格的内容</td>
</tr>
<table>
```

说明：文字与单元格的距离是以像素为单位的，默认设置为 0 像素。

6. 表格边框宽度

在 HTML 中，默认表格的边框宽度为零，即不显示表格的边框。如果要显示表格的边框，必须为表格指定边框宽度。在 HTML 中可以使用<table>元素的 border 属性来设置表格边框的宽度。其语法如下：

```
<table border="表格的边框宽度">
<tr>
<td>表格的内容</td>
</tr>
</table>
```

7. 表格边框颜色

在默认情况下，边框的颜色是灰色的，如果整个页面设置了特定的颜色，那么为了使表格和整个页面的色调协调一致，应该为表格的边框设置一个搭配的颜色，可以使用<table>元素的 bordercolor 属性来设置表格边框的颜色。其语法如下：

```
<table border="表格的边框宽度" bordercolor="边框颜色">
<tr>
<td>表格的内容</td>
</tr>
</table>
```

说明：边框颜色可以是颜色的英文名称，也可以是十六进制的颜色代码。需要注意的是，要想为边框设置颜色，首先必须为边框设置宽度，否则看不到效果。

8. 表格亮边框颜色

在 HTML 中，除了可以设置表格的边框颜色外，还可以将边框划分为亮边框和暗边框。亮边框就是光线照到的边框，一般认为表格外部的上边框和左边框以及内部的下侧和右侧边框为亮边框，可以通过<table>元素的 bordercolorlight 来设置表格亮边框。其语法如下：

```
<table border="表格的边框宽度" bordercolorlight="亮边框的颜色">
<tr>
<td>表格的内容</td>
</tr>
</table>
```

说明：在定义亮边框颜色的时候可以使用英文颜色名称，也可以使用十六进制颜色值。

9. 表格暗边框颜色

与表格的亮边框相对应的就是暗边框，一般认为是表格外部的下边框和右边框，可以通<table>元素的 bordercolordark 来设置表格暗边框，其语法如下：

```
<table border="表格的边框宽度" bordercolor dark="暗边框的颜色">
<tr>
<td>表格的内容</td>
</tr>
</table>
```

说明：在定义暗边框颜色的时候可以使用英文颜色名称，也可以使用十六进制颜色值。

11.4.3 表格行的对齐方式

表格都是由行组成的，行也可以设置对齐方式，通过对行的对齐方式进行设定，可以使表格更加整齐。表格行的对齐方式包括水平对齐方式和垂直对齐方式。

1. 垂直对齐方式

valign 属性可以设置行的垂直对齐方式，就是使行里面的内容都垂直对齐，其中默认为垂直居中对齐。垂直对齐方式的语法结构如下：

```
<table>
<tr valign="">
<td>表格的内容</td>
</tr>
</table>
```

说明：valign 属性有 3 个值，即 middle、top、bottom，分别表示居中对齐、居上对齐、居下对齐。

这 3 个属性值除了可以写在<tr>标签里面以外，还可以写在<td>标签里面。写在<td>标签里面用来控制每列内容的垂直对齐方式，其用法和写在<tr>标签里面一样。

2. 水平对齐方式

align 属性可以设置行的水平对齐，就是使行里面的内容都水平对齐，默认为水平居左对齐。水平对齐方式的语法结构如下：

```
<table>
<tr align="">
<td>表格的内容</td>
</tr>
</table>
```

说明：align 属性共有 3 个值，即 center、right、left，分别表示居中对齐、居右对齐、居左对齐。

11.4.4 行和列的合并

在一个表格里，有时候用户想编辑的表格中并不一定每行每列的行数和列数都是一样的，这时候就需要进行行和列的合并。

1. 列的合并

colspan 属性可以进行列的合并，就是把行中的某个单元格与其后的一个或多个单元格合并，其语法结构如下：

```
<table>
<tr>
<td colspan=所跨的列数>表格的内容</td>
</tr>
</table>
```

说明：设置的是单元格所跨的列数而不是像素数。

2. 行的合并

rowspan 属性可以进行行的合并，就是把单元格与其下的一个或几个单元格进行合并，其语法结构如下：

```
<table>
<tr>
<td rowspan=所跨的行数>表格的内容</td>
</tr>
</table>
```

说明：在这里设置的是单元格所跨的行数。

11.4.5 表格的结构

表格结构标记可以明确表格的结构，分出表格的头（表首）、身体（表主体）和尾（表尾），通过设置这些结构标记还可以分别对表首、表主体以及表尾的样式进行设置。

1. 表头

通常，一个表格的第 1 行都是用于说明本列数据含义的表头行，标签<thead>用于显示表格的表头内容，表头标签<thead>的使用可以让网页中过长的表格在打印时每页的最前面都显示出表头标签<thead>的内容。表头语法如下：

```
<thead>
<tr>
<td>单元格内的文字</td>
</tr>
</thead>
```

2. 主体

表格的主体就是表格真正要表达的内容和数据，一般占用表格的大部分内容，通过表主体标记<tbody>可以更好地划分表格的结构。表格主体部分的设置语法如下：

```
<tbooy>
```

```
<tr>
<td>单元格内的文字</td>
</tr>
</tbooy>
```

3. 表尾

表格的表尾主要用于标注表格的一些额外信息，例如内容的设计者、创建日期、总和等。在 HTML 中，表格的表尾标记是<tfoot>，可以让网页中过长的表格在打印时每页的最后面都显示出表尾标签<tfoot>的内容。其语法如下：

```
<tfoot>
<tr>
<td>单元格内的文字</td>
</tr>
</tfoot>
```

11.4.6 表格的标题

表格经常包括标题部分，在 HTML 4.01 中专门有一个<caption>元素用来设置表格的标题，不过由于表格的标题部分看上去与表格似乎是分离的两个部分，所以很多网页开发者都没有使用该元素的习惯，而是直接使用文本元素来设置表格标题。在默认情况下，表格的标题都显示在表格的上方，并且是居中显示。通常，<caption>元素都是紧跟在<table>之后的，但是它可以出现在<table>元素与<tr>元素之间的任何位置，其语法如下：

```
<caption>表格的标题文字</caption>
```

11.4.7 表格的嵌套

在实际应用中，表格并不是单一出现的，往往需要在表格内嵌套其他的表格来实现页面的整体布局，一般情况下需要使用一些可视化软件来实现布局，这样看起来比较直观，容易达到预期的目的，当然也可以通过直接输入代码实现。

11.5 框架基础

框架是一种划分浏览器窗口的特殊方式，通过使用框架可以将多个网页组合成一个页面显示在浏览器中。浏览器的各个页面之间相互独立，却又相互关联。用户在浏览这种页面的时候，当对其中某一个部分进行操作，如浏览、下载的时候，其他页面保持不变，这样的页面被称为框架结构的页面，也称为多窗口页面。

11.5.1 创建框架

实际上，框架对象本身也是一类窗口，它继承了窗口对象的所有特征，并拥有所有的属性和方法。使用框架最主要的目的就是创建链接的结构，最常见的框架结构就是将网站的

导航条作为一个单独的框架窗口。框架主要包含两个部分，即框架集和具体的框架文件。框架集<frameset>用来定义一个 HTML 文件为框架模式，并设定视窗如何分割。其实框架集就是存放框架结构的文件，也是访问框架文件的入口文件，其基本语法如下：

```
<frameset>
<frame src="">
<frame src="">
…
</frameset>
```

说明：其中每个<frame>标记中都会定义一个框架页面，也就是使用 src 属性定义一个真正显示内容的页面地址。在一个框架集文件内定义了几个<frame>标记，就表示它将页面划分了几个窗口。

11.5.2 分割窗口

框架结构最大的特点就是将一个单独的窗口分割成多个窗口，窗口的分割方式需要在框架集页面中设置。框架集是框架结构的基础，只有通过框架集，各个框架窗口才能结合起来组成一个真正的网页文件。因此，对于框架集的属性设置也是使用框架最基础的操作，而分割窗口则是基础中的基础。

1. 上下分割

既然是框架结构的页面，就要包含多个框架窗口，设置框架窗口的排列方式是必不可少的工作。上下分割窗口就是在窗口的水平方向上划分出几条分割线，将页面分成由上到下几个窗口，需要使用 rows 属性，其语法如下：

```
<frameset rows="各个窗口的高度">
<frame src=" ">
<frame src=" ">
…
</frameset>
```

说明：其中，rows 属性定义的是各个窗口的高度值，这些值可以是绝对的像素值，也可以是相对于整个页面的百分比，各窗口高度值之间使用逗号分隔开。

2. 左右分割

左右分割窗口就是在浏览器中沿垂直方向分割为几个窗口，这些窗口左右分布，其设置属性是 cols，使用的语法如下：

```
<frameset cols="各个窗口的宽度">
<frame src="">
<frame src="">
…
</frameset>
```

说明：cols 也可以取多个值，每个值表示一个框架窗口的水平宽度，它的单位可以是像

素，也可以是占浏览器的百分比。与垂直分割窗口相同，一般设定了几个 cols 的值就需要有几个框架窗口。

3. 窗口的嵌套

除了可以对窗口进行上下分割和左右分割外，还可以混合利用这两种模式，这就是窗口的嵌套。嵌套分割窗口就是在框架集中嵌套框架集，在一个浏览器页面内既有上下分割的窗口，又有左右分割的窗口。窗口嵌套的语法如下：

```
<frameset>
<frame src=" "/>
<frameset>
<frame src  =" "/>
<frame src  =" "/>
</frameset>
</frameset>
```

说明：框架集里起码要有一个框架页面嵌套多个框架集，在被嵌套的框架集中最少要有两个子框架页面，否则嵌套就没有意义了。

11.5.3 设置框架边框

1. 设置框架边框显示或隐藏

在默认情况下，框架的边框是显示出来的，有时候，用户希望隐藏框架结构中的框架分割线，这时可以通过设置 frameborder 实现。设置框架边框的语法如下：

```
<frameset frameborder="框架显示属性值">
<frame src="">
<frame src="">
…
</frameset>
```

说明：在这里框架的显示属性值只能取 0 或 1，取 0 表示不显示边框，取 1 表示显示边框，默认效果是取值为 1 的效果。

2. 设置框架边框宽度

不同浏览器中对框架边框的宽度显示不一致，在 HTML 4.01 中也没有规定用于设置框架边框宽度的属性，但是很多浏览器都支持在<frameset>标签里使用 border 属性来设置边框宽度。设置框架边框宽度的语法如下：

```
<frameset border="框架边框宽度">
<frame src=" ">
<frame src=" ">
</frameset>
```

说明：边框宽度的值以像素为单位。

3. 设置框架边框颜色

在默认情况下，框架的边框是灰色的，可以使用 bordercolor 属性将其设置为其他颜色，但 bordercolor 属性主要在 border 属性存在的时候才可以产生效果。其语法如下：

```
<frameset bordercolor = "颜色值">
<frame src = " ">
<frame src = " ">
</frameset>
```

说明：颜色值可以是颜色的英文单词，也可以是十六进制的颜色代码。

11.5.4 框架的属性

1. 设置框架滚动条显示

在默认情况下，当框架窗口的内容不足以在分割的区域内显示时会出现滚动条，以方便用户查看。滚动条会自动在各个无法完全显示的框架窗口内出现，使用 scrolling 属性可以分别对各个框架窗口的滚动条进行设置，其语法如下：

```
<frameset>
<frame src = " " scrolling = "属性值">
</frameset>
```

说明：在这里 scrolling 的属性值可以取值为 yes、no 或者 auto，其含义如表 11-3 所示。

表 11-3 scrolling 的属性值

属性值	具 体 含 义
yes	一直显示滚动条，无论页面是否完全显示
no	从来不显示滚动条，即使内容无法完全显示出来
auto	根据页面的长度自动调整，当页面无法完全显示的时候就显示滚动条。该属性值是 HTML 框架结构页面的默认效果

2. 固定框架

在默认情况下，框架窗口的大小是可以由用户自己来调整的，不过有些时候网页开发者不希望用户可以自己调整框架窗口大小而影响了网页效果。在 HTML 4.01 里允许通过 frame 元素的 noresize 属性来禁止浏览用户调整框架窗口大小。固定框架的语法如下：

```
<frameset>
<frame src = " " noresize>
</frameset>
```

说明：noresize 属性没有属性值，哪个框架窗口使用了 noresize 属性，该框架窗口的大小则不能由用户调整。

3. 不支持框架标记

框架结构虽然对页面导航很有效，但是有些浏览器并不支持框架页面，因此框架结构的

网页有时候无法正常显示，这就影响了用户的阅读。使用<noframes>标记可以使这些浏览器能够读取标记内的内容，而对于支持框架结构的浏览器来说，则会自动忽略其中的内容。<noframes>被称为不支持框架标记，其使用的语法如下：

```
<frame set>
<frame src="页面源文件地址" noresize>
<noframes>
…
</noframes>
</frameset>
```

说明：在这里，放置在标记<noframes>和</noframes>之间的内容就是在不支持框架的浏览器中所显示的内容。

11.5.5 在框架中使用链接

框架的链接和普通链接一样，都需要使用<a>标记。需要特别指出的是，当框架结构中的一个框架作为变换页面时需要为该框架窗口命名，然后根据名称对框架进行变换，这就需要使用 name 标记和 target 标记。其中，name 标记用来为窗口命名，而 target 标记用来引导链接页面在框架内打开。这个操作和超链接中的目标属性是非常相似的。

11.5.6 浮动框架

在框架结构中，除了固定的分割方式之外，还有一种框架窗口可以作为一个页面元素添加到普通页面中，这样框架窗口可以位于页面的各个位置，这种框架结构被称为浮动框架。浮动框架带有许多属性，通过这些属性可以调整框架页面的样式以及它在页面中的布局，其属性如表 11-4 所示。

表 11-4 浮动框架的属性

属　　性	具体含义
src	浮动框架页面的源文件地址
width	浮动框架在页面中显示的宽度
height	浮动框架在页面中显示的高度
align	浮动框架页面在浏览器中的对齐方式，可以为左对齐、右对齐或居中对齐
name	设定框架页面的名称
marginwidth	设置框架边缘宽度
marginheight	设置框架边缘高度
scrolling	设定浮动框架页面内是否显示滚动条
frameborder	设定浮动框架的边框

1. 插入浮动框架

在 HTML 里可以使用<iframe>标签创建浮动框架。<iframe>标签与<img>标签一样，可以放在<body>标签之内的任何一个位置，<iframe>标签里的 src 属性可以用来指定浮动框架载入的文档 URL。插入浮动框架语法如下：

```
<iframe src="页面源文件地址">
</iframe>
```

2. 浮动框架大小

在普通框架结构中，由于框架就是整个浏览器窗口，因此不需要设置大小，但是浮动框架是插入到普通 HTML 页面中的一个元素，可以使用 width 和 height 属性调整大小，其语法如下：

```
<iframe src="页面源文件地址" width="窗口宽度值" height="窗口高度值">
框架内容
</iframe>
```

说明：在这里，width 属性设置的是框架窗口的宽度，height 属性设置的是框架窗口的高度，它们的单位都是像素。

3. 浮动框架对齐方式

对于浮动框架来说，在默认情况下是左对齐的，如果不想让浮动框架左对齐，可以使用 align 属性对框架的对齐方式进行更改，其语法如下：

```
<iframe src="" align="left/center/right">
框架内容
</iframe>
```

说明：与设置其他页面元素的对齐方式类似，浮动框架的对齐方式包括 left、center 和 right，分别表示左对齐、居中对齐和右对齐。

4. 浮动框架页面的链接

除了普通的框架结构外，浮动框架也可以制作页面之间的链接。在一个浮动窗口内可以链接多个页面，首先为浮动框架窗口定义名称，然后将链接文字的目标页面打开方式设置为定义的窗口名称，即将超链接的 target 属性设置为浮动窗口的名称，这样，当运行程序的时候就会在浮动窗口打开链接的目标页面了。

11.6 表单基础

大家在现实生活中需要填很多表，例如入学申请表、健康体检表等，如果把这些表格放在网上，就是 HTML 中的表单。表单是实现动态页面的一种主要的外在形式，也就是说表单是进行用户和浏览器交互的重要手段。

11.6.1 添加表单

表单可以用来收集用户在客户端提交的各种信息，例如在网站登录或注册时进行的键盘和鼠标操作都是通过表单作为载体传递给服务器的。表单其实是页面中的一个特定的区域，由<form>标记和</form>标记定义，所有的表单元素都要在这对标记之间才有效。

表单的基本语法如下：

```
<form>
设置各种表单元素
</form>
```

1. 链接跳转

action属性用来设置链接跳转，也就是在提交表单内容的时候按照链接地址跳转到相应的页面进行处理。由于action属性用来控制整个表单的提交内容，所以action属性要写在<form>标签里面。其语法结构如下：

```
<form action="链接跳转的地址">
设置各种表单元素
</form>
```

说明：链接跳转的地址除了可以是绝对地址和相对地址外，还可以是一些其他的地址形式，如表单中没有任何表单元素，那么这个表单传递给处理程序的内容就是空的。

如果省略action属性，默认为提交到本页，也就是说本页即为接收并处理表单的程序。可以接收并处理表单的程序有很多，常用的有ASP、ASPX、JSP、PHP等。

2. 链接跳转方式

在设置了链接跳转action以后，还需要设置当链接跳转时所使用的跳转方法，可以通过method属性进行设置，它决定了表单中已收集数据是用什么样的方法发送到服务器的。其语法结构如下：

```
<form method="表单的链接跳转方法">
表单元素
</form>
```

表单的链接跳转方式一般可以设置为get和post两种，其具体含义如表11-5所示。

表11-5　HTML中的表单链接跳转方式

传送方法	具体含义	注意事项
get	表单数据会被视为cgi或asp的参数发送，用户输入的数据会附加在URL之后，由用户端直接发送至服务器，是method属性的默认值	其速度较快，但数据不能够太长。如果信息超过8192个字符，则会被截去。另外，该方法不具有保密性
post	表单数据与URL分开，将数据写在表单主体内发送	没有字符长度的限制，可以发送较长的信息，但速度相对较慢

3. 表单名称

表单名称标记主要是为了区分各个表单，因为有时候在一个页面中可能会有多个表单，或者在一个表单处理程序中需要处理多个页面的表单，这时候表单名称就显得尤其重要了。表单名称的标记是name，其语法如下：

```
<form name = "表单名称">
表单元素
</form>
```

说明：name 属性的写法和超链接里 name 属性的写法一样。

11.6.2 输入标签

输入标签<input>是使用最广泛的表单控件元素，用于定义输入域的开始。<input>标签是单标签，所以在使用时要为<input>标签加上“/”号实现标签的闭合。<input>标签必须嵌套在表单标记中使用。其语法结构如下：

```
<form>
<input type = " "/>
</form>
```

说明：type 属性的值有很多不同的选择对应不同的输入方式。

1. 文本框

文本框用来在输入框中输入任何类型的数字、文本以及字母等，输入的内容单行显示在页面中，可以通过 type="text"进行设置。其语法如下：

```
<form>
<input type = "text">
</form>
```

2. 密码框

密码框是用来输入密码的，可以通过 type="password"进行设置。在设置以后，在密码框中输入的内容就会变成小黑点或者是“ * ”号，可以用来保护密码不被第三者看见。其语法结构如下：

```
<form>
<input type = "password">
</form>
```

3. 单选框

单选框指只能选择其中一项的选项框，就像很多表单中的“性别”选项一样，要么是男，要么是女，不可能同时是男又是女。因此，可以将性别这一单元设置成单选框，让用户选择，选中的选项会以圆点显示。单选框可以通过 type="radio"进行设置，其语法如下：

```
<form>
<input type = "radio">
</form>
```

4. 复选框

复选框与单选框类似，可以是一个单独的复选框，也可以是多个复选框组成的复选框

组。复选框可以让用户同时在表单中选择或取消多个选择项目，在浏览器中通常表现为一个小方框，当用户选中复选框时，会在小方框里打一个勾，如果用户没有选中该选项或取消该复选框，小方框里为空。复选框可以通过 type="checkbox"进行设置，其语法如下：

```
<form>
<input type="checkbox" name="名称">
</form>
```

说明：同一组复选框，name 属性必须相同。

5. 提交按钮

提交按钮是把表单里的信息提交到指定的数据库或者其他地方，可以通过 type="submit"进行设置，其语法结构如下：

```
<form>
<input type="submit" name="名称" value="">
</form>
```

说明：value 属性里填写的是按钮上面出现的文字，以表示该按钮是提交按钮，可以是中文，也可以是英文。

6. 重置按钮

重置按钮的作用是将表单中的用户输入清除，将内容恢复到初始状态。重置按钮可以通过 type="reset"进行设置，其语法如下：

```
<form>
<input type="reset" name="名称" value="">
</form>
```

说明：value 属性里填写的是按钮上面出现的文字，以显示该按钮是重置按钮，可以是中文，也可以是英文。

7. 图像按钮

图像按钮是指将页面中的按钮使用图片显示外观，这样的图片可以具有按钮的功能，而且页面更加美观。图像按钮可以通过 type=" image"设置，其语法如下：

```
<form>
<input type="image" src="图像源文件">
</form>
```

说明：由于在这里需要使用图像，因此和插入图像一样，需要使用 src 属性正确设置图像的源文件地址。

8. 文件域

文件域可以让用户选择存储在本地计算机上的文件，通常用于在文件上传到服务器时进行文件的选择。文件域在浏览器里的显示为一个文本框与一个按钮，通常按钮上会显示

"浏览"字样。这两个组件是同时出现在网页中的，当单击【浏览】按钮时，会弹出一个【选择文件】对话框，在该对话框中选择文件之后单击【打开】按钮，就会在文本框中自动输入该文件在本地的绝对路径。文件域可以通过 type="file"设置，其语法如下：

```
<form>
<input type="file" name="名称">
</form>
```

9. 隐藏域

在前面章节中介绍的表单元素在浏览器里都是可以看到的，在 HTML 中还有一种表单元素在浏览器里是看不到的，这种表单元素称为隐藏域或隐藏框，隐藏域的作用是在表单里放入一个不希望被用户看到或用户没有必要看到的内容，而这些内容往往是在提交表单时服务器或脚本需要获取的内容。隐藏域可以通过 type="hidden"设置，其语法如下：

```
<form>
<input type="hidden">
</form>
```

11.6.3 下拉列表

下拉列表是一个下拉式的列表或者带有滚动条的列表，用户可以在列表中选择一个选项。创建下拉列表需要用到两个元素，首先要用到 select 元素，用于标记下拉列表开始，然后要用到 option 元素，用于创建下拉列表中的项目。如果一个下拉列表中有多个可选项目，只需要重复使用 option 元素。下拉列表的语法如下：

```
<select name="名称">
<option value=" ">选项内容</option>
<option value=" ">选项内容</option>
<option value=" ">选项内容</option>
…
</select>
```

说明：与单选框和复选框一样，select 元素也需要使用 name 元素告诉服务器该表单的名称。在提交表单时，服务器通过 select 元素的 name 属性值获得该下拉列表框里的选项值，而 select 元素的选项值要在 option 元素里通过 value 属性设置。

11.6.4 文本域

单行文本框只是在网页中显示一个只能输入一行文字的文本框，如果用户有大量的文字，尤其是分段的多行文字，在单行文本框里是无法输入的。HTML 中提供了一个<textarea>元素，使用该元素可以在网页里创建一个文本域。在文本域里可以显示多行文字，也可以输入多行文字，在很大程度上方便了用户输入和查看文字。文本域的语法结构如下：

```
<textarea>
输入的内容
</textarea>
```

11.7 多媒体和滚动文字

在现在的网页中，只有文字和图片是完全不够的，还要有动画、声音等的加入，这样整个页面才能更加吸引人。在 HTML 中就提供了插入各种多媒体元素、滚动字幕的功能。

11.7.1 多媒体元素

1. 插入多媒体元素

在 HTML 中添加多媒体元素的标记是<embed>，这里的多媒体元素包括声音和动画两种。

也就是说，除了可以设置背景音乐之外，还可以为页面添加声音和动画文件，使页面动感十足。在页面中添加多媒体文件同样要设置源文件，src 属性是必不可少的。另外，由于多媒体文件在播放时需要一定的空间，因此要为其设置大小。可以说，要在页面中添加多媒体元素，仅使用<embed>标记是不够的，其语法如下：

```
<embed src="源文件地址" width="多媒体显示的宽度" height="多媒体显示的高度"></embed>
```

说明：一般情况下，不在标记<embed>和</embed>之间添加其他的内容。

2. 循环播放

在默认情况下，多媒体文件播放一次以后就会自动停止。如果希望该文件循环播放，则需要使用 loop 属性进行设置，其语法如下：

```
<embed src="源文件地址" width="多媒体显示的宽度" height="多媒体显示的高度" loop=循环
播放></embed>
```

说明：一般情况下，在设置循环播放属性的时候，需要将 autostart 设置为 true。

3. 自动播放

通过<embed>元素中的 autostart 属性可以设置打开网页时背景音乐是自动播放，其语法如下：

```
<embed src="源文件地址" width="多媒体显示的宽度" height="多媒体显示的高度" autostart
=是否自动播放></embed>
```

说明：autostart 属性可以取值为 true 和 false，其中，true 表示自动播放，而 false 表示需要手动播放。

4. 隐藏多媒体元素

有时候，用户希望在网页中只听到多媒体文件的播放声音，而看不见多媒体文件，这时可以使用<embed>元素中的 hidden 属性来隐藏多媒体的面板，这样在播放的时候就只能

听到声音，而看不到画面。其语法如下：

```
<embed src="源文件地址" width="多媒体显示的宽度" height="多媒体显示的高度" autostart=
true hidden=隐藏值></embed>
```

说明：hidden 的值可以取 true 或 false，取 true 表示隐藏面板，反之显示面板。如果设置了 hidden 的值为 true，则要将 autostart 的值设置为 true，否则用户无法播放多媒体，也就失去了添加该多媒体文件的意义。

11.7.2 插入背景音乐

除了可以添加视频外，还可以通过<bgsound>元素为网页添加背景音乐。和图像标记一样，<bgsound>元素的源文件地址属性 src 是必需的，一般情况下，背景音乐要添加在页面主体的开始位置。添加背景音乐的语法如下：

```
<bgsound src="源文件地址">
```

说明：作为背景音乐的可以是 MP3 音乐文件，也可以是其他声音文件，而在网络中应用最广泛的是 MIDI 声音文件。

11.7.3 滚动字幕

滚动字幕也可以称为一种多媒体元素，只不过这种类型的多媒体比较简单，实现的效果比较单一，在一些时尚感要求较低的站点中可以使用动态文字来增加页面的动感。

1. 添加滚动字幕

滚动字幕是指在网页中会上下活动或左右活动的字幕，可以是文字也可以是图片，可以通过<marquee>标记来设置。添加滚动字幕的语法如下：

```
<marquee>要进行滚动的文字</marquee>
```

说明：标记<marquee>和</marquee>之间的文字在页面中滚动显示，默认情况下是从右向左滚动。

2. 滚动方向

滚动方向是指文字从哪个方向开始滚动。在默认情况下文字是从右向左滚动的，使用 direction 属性可以调整文字的滚动方向。设置文字滚动方向的语法如下：

```
<marquee direction="left | right | up | down">
要进行滚动的文字
</marquee>
```

说明：direction 属性有 4 个值，即 left、right、up、down，分别用来设置字幕从右向左滚动、从左向右滚动、从下向上读动、从上向下滚动。

3. 滚动方式

文字的滚动方式主要包括循环滚动、来回滚动以及只滚动一次就停止。设置文字的滚

动方式需要使用 behavior 属性,其语法如下:

```
<marquee behavior = "滚动方式">要进行滚动的文字</marquee>
```

在这里,滚动方式的取值如表 11-6 所示。

表 11-6　滚动方式的取值

滚动方式	具 体 含 义
scroll	循环滚动,即不停地按照设置的方向滚动
slide	只滚动一次就停止
alternate	交替滚动,即按照设置的方向和反方向来回滚动。例如设置为 left,则先向左滚动,滚动到页面左端后再向右滚动

4. 滚动字幕的背景颜色

在页面中为了突出滚动文字,可以使用<marquee>元素中的 bgcolor 属性为其添加背景颜色。其语法如下:

```
<marquee bgcolor = "颜色值">要进行滚动的文字</marquee>
```

5. 滚动速度

在 HTML 中,可以通过<marquee>元素中的 scrollamount 属性调整文字的滚动速度,滚动速度也可以看成滚动距离,也就是每滚动一下文字向前移动的像素数。其设置语法如下:

```
<marquee scrollamount = "滚动速度">要进行滚动的文字</marquee>
```

说明:滚动速度的值越大,滚动越快。

6. 滚动延迟

滚动延迟是指在每一次滚动之间设置一定的时间间隔,即滚动一次后停止一段时间再进行下一次滚动。设置滚动延迟的语法如下:

```
<marquee scrolldelay = "延迟时间">要进行滚动的文字</marquee>
```

说明:延迟时间的单位是毫秒,即设置为 100 表示延迟 0.1s,设置为 1000 表示延迟 1s。

7. 滚动次数

除了可以通过 behavior 设置文字的滚动方式外,还可以通过 loop 属性设置文字具体的滚动循环次数。其语法如下:

```
<marquee loop = "循环次数">要进行滚动的文字</marquee>
```

说明:在这里 loop 表示循环次数,如果设置为 10,则表示文字在屏幕中滚动 10 个循环后结束滚动。

8. 滚动字幕空白空间

滚动字幕空白空间是指在滚动文字区域周围的空间，默认情况下，滚动区域是沿着页面的边缘滚动的。如果没有使用段落标记等将其分隔，它和页面中的其他元素是紧紧相连的。通过 hspace 属性和 vspace 属性可以设置文字区域的水平和垂直空间，其语法如下：

```
<marquee hspace="水平空间" vspace="垂直空间">要进行滚动的文字</marquee>
```

说明：水平空间和垂直空间的单位都是像素。

9. 设置鼠标经过效果

onmouseover 属性用来控制鼠标滑过滚动字幕时停止滚动的效果，onmouseout 属性用来控制鼠标移出滚动字幕区域时字幕开始滚动的效果，这两个属性必须同时定义。其语法如下：

```
<marquee onmouseout="this.start()" onmouseover="this.stop()">要进行滚动的文字</marquee>
```

说明：onmouseout＝"this.start()"用来设置鼠标移出该区域时继续滚动，onmouseover＝"this.stop()"用来设置鼠标移入该区域时停止滚动。这两个属性同时使用，才可以使鼠标滑过滚动字幕时停止滚动，而当鼠标移开滚动字幕时又开始滚动。

11.8 综合应用

11.8.1 蝴蝶之恋

本例中先应用 Photoshop 处理图片，然后应用 HTML 中的文字、图片、表格等制作一个蝴蝶之恋的主页。

（1）HTML 网页文件的代码如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/
xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title>蝴蝶之恋</title>
<style type="text/css">
.aa{
    text-align: center;
}
body{
    background-color: #93F;
}
a:link{
```

```
    color: #FFF;
}
.bb{
    color: #900;
}
</style>
</head>

<body>
<table width="800" border="0" align="center" cellpadding="0" cellspacing="0">
  <tr>
    <td class="bb"><span class="aa">
      <marquee>      蝴蝶之恋      </marquee>
    </td>
  </tr>
  <tr>
    <td><hr/></td>
  </tr>
  <tr>
    <td><table width="200" border="1" align="center" cellspacing="5">
      <tr>
        <td><img src="1.jpg" width="265" height="240" /></td>
        <td><img src="2.jpg" width="265" height="240" /></td>
      </tr>
      <tr>
        <td><img src="3.jpg" width="265" height="240" /></td>
        <td><img src="4.jpg" width="265" height="240" /></td>
      </tr>
    </table></td>
  </tr>
  <tr>
    <td class="aa">::::请你单击上面的图片进入本站首页::::<br />
      BEST VIEW   800*600</td>
  </tr>
  <tr>
    <td><hr /></td>
  </tr>
  <tr>
    <td class="aa">|<a href="#">本站首页</a>|<a href="shuijiao1.html">心情小筑</a
>|<a href="shuijiao2.html">蝴蝶之恋</a>|<a href="#">蝴蝶随笔</a>|<a href="#">关
于我们</a>|<a href="#">和我联系</a>|</td>
  </tr>
</table>
</center>
</body>
</html>
```

(2) 代码在浏览器中运行的效果如图 11-2 所示。

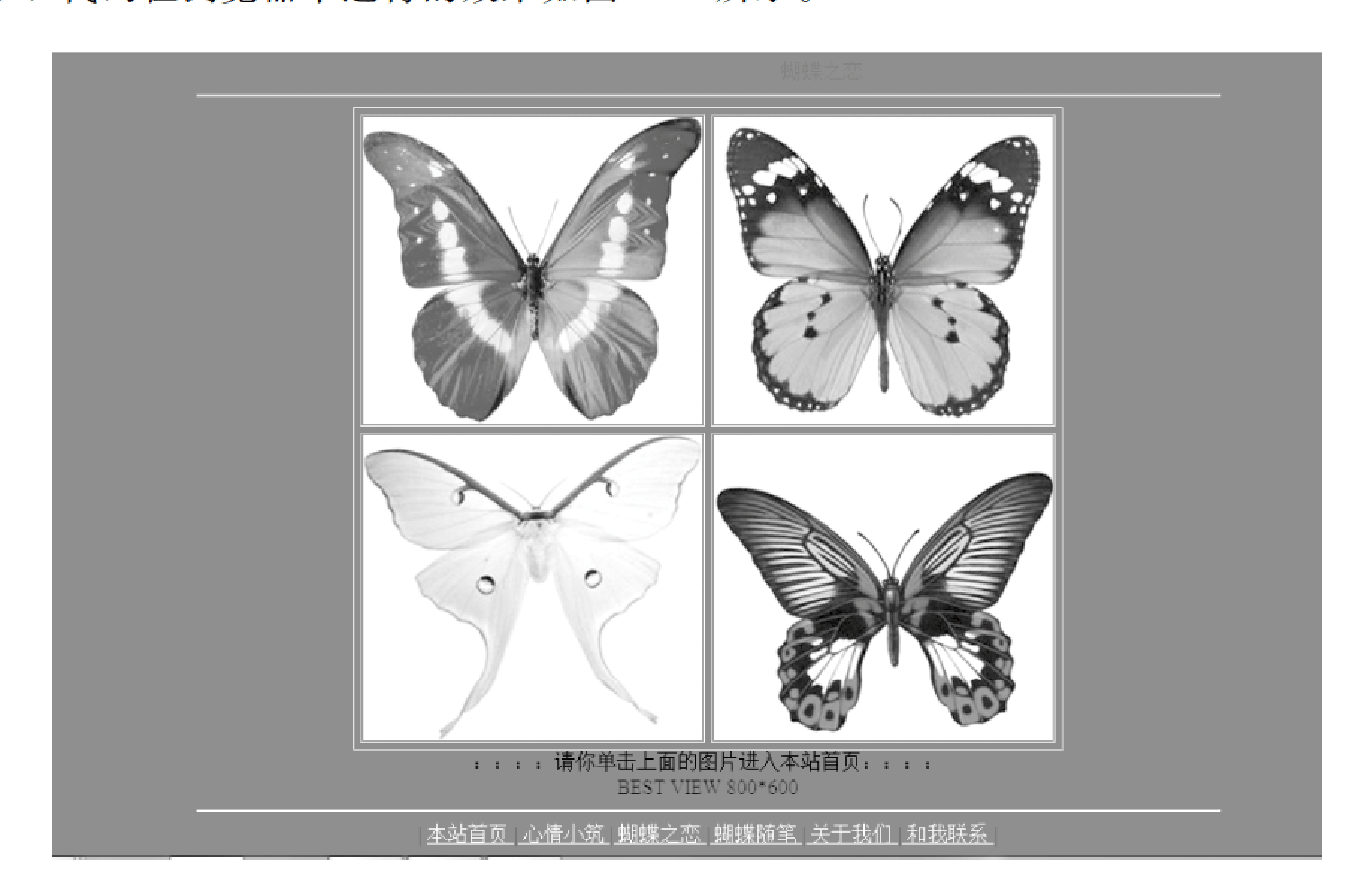


图 11-2　蝴蝶之恋网页效果

11.8.2　个人主页

本例中先应用 Photoshop 的切片工具，再应用 HTML 中的 iframe 制作一个个人主页。

(1) 启动 Photoshop CS6，打开 wangye.jpe 图片，利用切片工具将图片切割，然后选择【文件】|【存储为 Web 所用格式】命令，将文件存为 HTML 和图像，得到一个名为"11.8.1.html"的文件和一个名为"img"的文件夹。

(2) 启动 Dreaweaver CS6，打开"11.8.1.html"文件，对其进行相应修改，代码如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/
xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>个人主页</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<body bgcolor="#FFFFFF" leftmargin="0" topmargin="0" marginwidth="0" marginheight="0">
<!-- Save for Web Slices (wangye.jpg) -->
<table id="_01" width="778" height="654" border="0" cellpadding="0" cellspacing="0">
<tr>
    <td colspan="2">
        <img src="images/wangye_01.gif" width="376" height="158" alt=""></td>
    <td rowspan="3">
        <img src="images/wangye_02.gif" width="402" height="654" alt=""></td>
</tr>
<tr>
    <td rowspan="2">
        <img src="images/wangye_03.gif" width="66" height="496" alt=""></td>
    <td background="images/wangye_04.gif">
```

```
        <iframe src = "first.html" width = "100%" height = "270" name = "kuangjia" noresize
marginwidth = "0" marginheight = "0" frameborder = "0" framespacing = "0" scrolling = "no"
allowtransparency = "true" style = "filter: chroma(color = #ffa7da)"></iframe></td>
</tr>
<tr>
    <td>
        <img src = "images/wangye_05.gif" width = "310" height = "204" alt = ""></td>
</tr>
</table>
<!-- End Save for Web Slices -->
</body>
</html>
```

(3) 框架中的滚动文字是一个单独的网页文件，文件名为“first.html”，其代码如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/
xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns = "http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv = "Content-Type" content = "text/html; charset = gb2312" />
<title>滚动文字</title>
<style type = "text/css">
<!--
body {
     background-color:transparent;
}
.style1 {
font-size: 12px;
color: #333333;
}
.style2 {color: #333333}
-->
</style>
</head>
<body>
<span class = "style2">
<marquee direction = "up" scrollamount = "1" height = "290">
</span>
<p align = "center" class = "style1">
轻轻的我走了,<br />
正如我轻轻的来; <br />
我轻轻的招手,<br />
作别西天的云彩。</p>
<p align = "center" class = "style1">
那河畔的金柳, <br />
是夕阳中的新娘; <br />
波光里的艳影, <br />
在我的心头荡漾。</p>
<p align = "center" class = "style1">
软泥上的青荇, <br />
油油的在水底招摇; <br />
在康河的柔波里, <br />
我甘心做一条水草!
</p>
```

```
<p align="center" class="style1">
那榆荫下的一潭,<br />
不是清泉,<br />
是天上虹;<br />
揉碎在浮藻间,<br />
沉淀着彩虹似的梦。</p>
<p align="center" class="style1">
寻梦?撑一支长篙,<br />
向青草更青处漫溯;<br />
满载一船星辉,<br />
在星辉斑斓里放歌。</p>
<p align="center" class="style1">
但我不能放歌,<br />
悄悄是别离的笙箫;<br />
夏虫也为我沉默,<br />
沉默是今晚的康桥!</p>
<p align="center" class="style1">
悄悄的我走了,<br />
正如我悄悄的来;<br />
我挥一挥衣袖,<br />
不带走一片云彩。</p>
</body>
</html>
```

(4)代码在浏览器中运行的效果如图 11-3 所示。

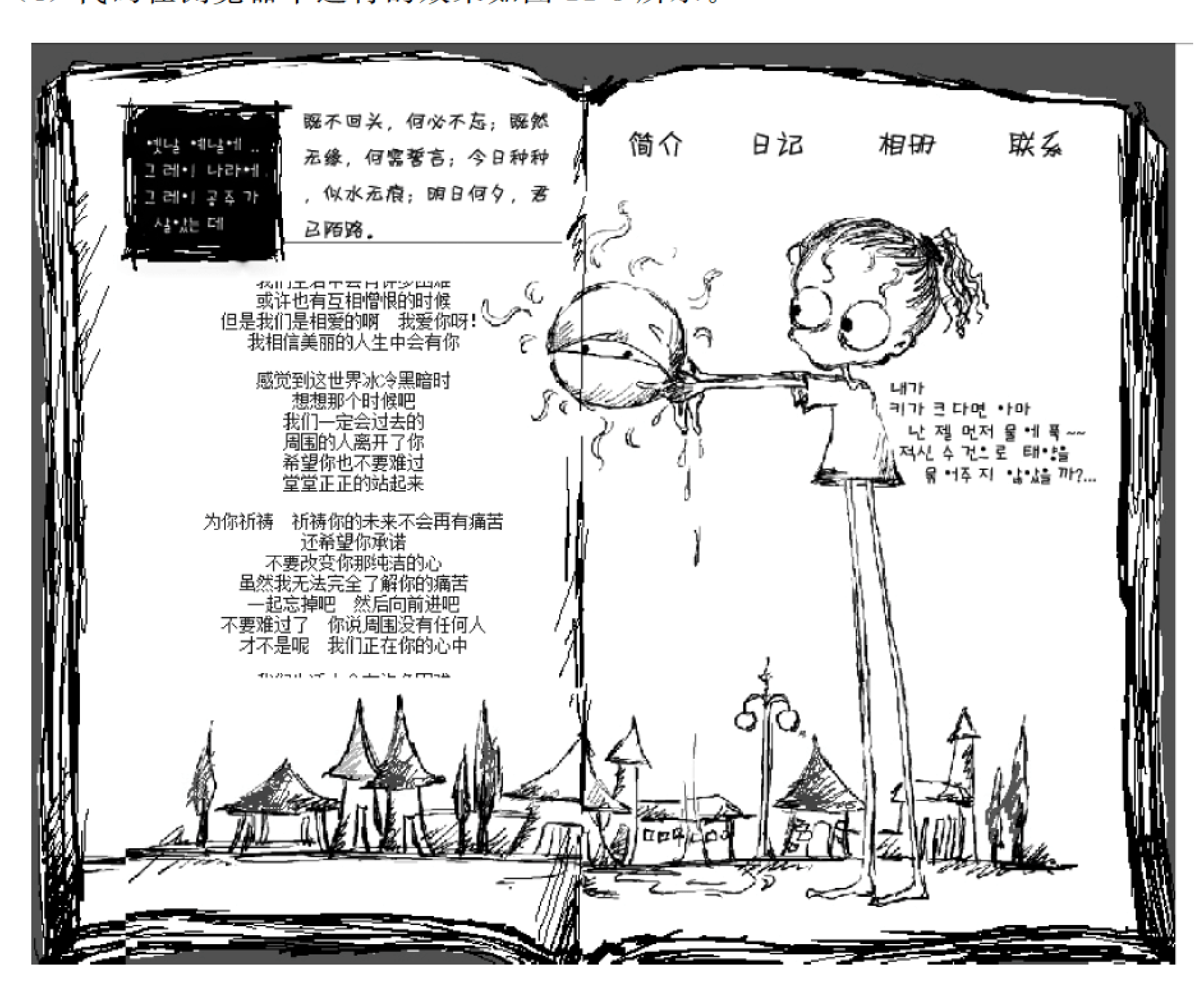


图 11-3 个人主页效果

第12章 CSS与DIV

本章学习目标：

✍了解 CSS 基本规则。
✍掌握利用 CSS 美化网页的方法。
✍掌握利用 CSS 布局网页的方法。

12.1 CSS 基础

CSS 是 Cascading Style Sheets(层叠样式表)的缩写。它的作用是定义网页的外观(例如字体、颜色等),控制网页的布局,它也可以和 JavaScript 等浏览器端脚本语言结合使用制作出许多动态的效果。

12.1.1 CSS 样式表的设置方法

当浏览器读取样式表时要依照文本格式来读,这里介绍 4 种在页面中插入样式表的方法,即内联样式表、内部样式表、外部样式表和引用多个样式表。

1. 内联样式表

写在标签内的样式称为内联样式,在标签内编写的样式能影响的范围最小,仅仅影响该标签内的文字,另一个标签内的文字将无法显示该标签所定义的样式。设置内联样式的语法如下：

```
<标签名 style = "样式属性 1:属性值 1;样式属性 2:属性值 2;...">
```

说明：内联样式可以用在<body>标签内的所有子标签,包括<body>在内,但不能用在<body>标签之外的标签上,如<head>、<title>、<html>等。<script>标签虽然也可以放在<body>标签内,但不能使用样式。

2. 内部样式表

在标签内设置样式可以影响该标签内的文字,但其影响范围太小。如果一个 HTML 文档里有多个相同样式的标签,使用内联样式要为每一个标签都设置一次,不能体现出 CSS 的强大功能。

在HTML文件里使用<style>标签可以设置影响整个文档的样式，这种方式称为内部样式。其语法如下：

```
<style type="text/css">
<!--
选择符1{样式属性:属性值;样式属性:属性值;…}
选择符2{样式属性:属性值;样式属性:属性值;…}
选择符3{样式属性:属性值;样式属性:属性值;…}
…
-->
</style>
```

说明：<style>标签用于声明样式，type属性声明样式元素是以CSS的语法来定义的。

<!--…-->标签用于隐藏代码，当某些浏览器不支持CSS时，使用该标签可以让浏览器忽略其中的代码，避免出现错误。

3. 外部样式表

外部链接CSS是将样式表以单独的文件存放，让网站中的所有网页均可引用此样式，以降低维护的人力成本，并可让网站拥有一致的风格。这种设置方式是把样式表单独保存为一个文件，然后在页面中用<link>标记链接，而这个<link>标记必须放到页面的<head>区域内。其基本语法如下：

```
<link rel="stylesheet" type="text/css" href="样式表源文件地址">
```

说明：href属性里的外部样式文件的地址填写方法和超链接的链接地址写法一样。rel="stylesheet"表示告诉浏览器连接的是一个样式表文件，是固定格式。type="text/css"表示传输的文本类型为样式表类型文件，这也是固定格式。

一个外部样式表文件可以应用于多个页面。当改变这个样式表文件时，所有页面的样式都随之改变。样式表文件可以用任何文本编辑器(例如记事本)打开并编辑，一般样式表文件的扩展名为.css，其内容是定义的样式，不包含HTML标记。

4. 使用@import引用外部样式表

与<link>标签类似，使用@import也能引用外部样式，不过@import只能在<style>标签内使用，而且必须放在其他CSS样式之前。@import的语法如下：

```
@import url(外部样式地址);
```

说明：url为关键字，不能随便更改；外部样式地址指的是外部样式的URL，可以是绝对URL，也可以是相对URL。@import除了语法和所在位置与<link>标签不同以外，其他的使用方法与效果都是一样的。

使用@import引用外部样式，在该语句的最后一定要有分号，否则引用外部样式将会失败。

12.1.2 选择符

CSS 最大的作用就是能将一种样式加载到多个标签上,方便开发者管理与更改。CSS 通过选择符实现哪些标签元素使用样式,哪些标签元素不使用样式,也可以通过选择符指定标签元素使用哪个样式。

1. 类型选择符

类型选择符(Type Selectors)是以文档中的对象(如 Element)名作为选择符名,因此类型选择符可以使一个元素从原有的样式转变成另一种样式。类型选择符的语法如下:

```
E{样式属性:属性值;样式属性:属性值;...}
```

说明:E 为文档中的元素,如果该文档是 XML 文档,则 E 为 XML 文档中所声明的元素;如果该文档是 HTML 文档,则 E 为 HTML 的元素,如 p、h1、hr、img 等。

当在 CSS 里声明了某个元素的类型选择符之后,在该文档中所有被声明的元素都将被赋予该样式。

2. 类选择符

1) 类选择符可以与元素配合使用

当类选择符与元素配合使用时,类选择符的语法如下:

```
E.classname{样式属性:属性值;样式属性:属性值;...}
```

说明:E 为元素名称,classname 是标签用于选择样式所用的名称,只有 E 元素才能选择是否使用 classname。

2) 独立于元素的类选择符

类选择符可以与元素配合使用,也可以独立于元素使用,当类选择符独立于元素使用时,类选择符的语法如下:

```
.classname{样式属性:属性值;样式属性.属性值;...}
```

说明:classname 声明样式,所有的元素都可以通过 class="classname"来引用该样式。

3. ID 选择符

ID 选择符(ID Selectors)的使用方法与类型选择符和类选择符有点相似,其语法如下:

```
#idname{样式属性:属性值;样式属性:属性值;...}
```

说明:idname 是对应这个 id 的属性值,ID 选择符只是针对网页中某一个元素的,这个元素可以随便是什么元素,但其 id 的属性值必须是 ID 选择符的名字。

4. 包含选择符

在 CSS 的选择符里有一种选择符叫包含选择符(Descendant Selectors),在包含选择符里可以为一个特定的结构创建样式。例如,可以创建一个超链接的样式,但该样式只有在超

链接包含在<p>标签内时才有作用。包含选择符的语法如下：

```
E1 E2{样式属性:属性值;样式属性:属性值;...}
```

说明：只有包含在 E1 里的 E2 选择符才会应用其后的属性值。包含选择符不仅可以包含类型选择符，也可以包含类选择符。

5. 选择符分组

在一个 HTML 文档里，有可能多个标签使用同一种样式，即可以把相同样式的选择符放在一个组内。选择符分组的方式如下：

```
E1,E2,E3...{样式属性:属性值;样式属性:属性值;...}
```

说明：E1、E2、E3 是不同的选择符，它们将使用后面的同一样式。

6. 适用选择符

（1）通用选择符（Universal Selectors）的语法如下：

```
*{样式属性:属性值;样式属性:属性值;...}
```

说明：* 代表所有，即所有的标签都使用该样式。

（2）让某一标签下的所有标签使用同一个样式，此时通用选择符的语法如下：

```
E *{样式属性:属性值;样式属性:属性值;...}
```

说明：让 E 标签下的所有标签都使用该样式。

7. 子选择符

从某个程度上来说，子选择符（Child Selectors）与包含选择符很相似，但必须从父级标签指定子标签，其语法如下：

```
E1 > E2{样式属性:属性值;样式属性:属性值;...}
```

说明：其中 E1 为 E2 的父级标签。

8. 相邻选择符

相邻选择符是一个比较有意思的选择符，该选择符作用于兄弟标签，但只能作用在相邻的两个兄弟标签之间，其语法如下：

```
E1 + E2{样式属性:属性值;样式属性:属性值;...}
```

说明：E2 为紧跟在 E1 之后的兄弟标签，并且样式只能作用在 E2 上，不能作用在 E1 上。

9. 属性选择符

前面所介绍的选择符有针对元素声明的（如类型选择符）、有针对 id 值声明的（ID 选择符），在 CSS 中还有一种选择符，是针对元素中的不同属性声明的，这种选择符就是属性选择符（Attribute Selectors）。属性选择符的语法如下：

```
E[属性]{样式属性:属性值;样式属性:属性值;...}
E[属性=属性值]{样式属性:属性值;样式属性:属性值;...}
E[属性~=属性值]{样式属性:属性值;样式属性:属性值;...}
E[属性|=属性值]{样式属性:属性值;样式属性:属性值; ...}
```

如上所示，属性选择符有4种表达方式，不同的表达方式所代表的意思不一样。

1）第一种表达方法

属性选择符的第一种表达方式为：

```
E[属性]{样式属性:属性值;样式属性:属性值;...}
```

说明：这种属性选择符的意思是只要元素中包含该属性，就可以使用其后面的样式。

2）第二种表达方式

属性选择符的第二种表达方式为：

```
E[属性=属性值]{样式属性:属性值;样式属性:属性值;...}
```

说明：这种属性选择符的意思是元素中包含该属性，并且该属性的值与设定的值相同，那么就可以使用其后面的样式。

3）第三种表达方式

属性选择符的第三种表达方式为：

```
E[属性~=属性值]{样式属性:属性值;样式属性:属性值;...}
```

说明："~="符号就像是约等于，只要E标签的属性中有一个单词属于属性选择符中的属性值，就可以使用其后面的样式。

4）第四种表达方式

最后一种属性选择符的表达方式为：

```
E[属性|=属性值]{样式属性:属性值;样式属性:属性值;...}
```

说明：这种属性选择符比上一种属性选择符所能匹配的范围要小得多，上一种属性选择符所能匹配的是单词，而这一种属性选择符只可以匹配以连字符(-)分隔的字符串，并且只能是以属性值开头的元素。

12.1.3 伪类和伪元素

伪类(pseudo-class)和伪元素(pseudo-element)可以说是HTML文档中并不实际存在的类和元素：伪类通常指某些元素的某个状态，例如超链接元素存在4个状态，即未访问过的链接、已访问过的链接、鼠标经过时的链接和鼠标单击时的链接。伪元素通常指某个对象中某个元素的状态，例如一行文字中第一个字符的样式等。

1. 超链接的伪类

伪类最开始被提出时可以说完全是因为超链接，在CSS 1中只有3个伪类，即link、:visited和:active，这3个伪类都是作用在超链接上的，分别代表超链接的3个状态，即未访问过的URL、已访问过的URL、正在单击的超链接。后来，在CSS 2中对鼠标经过超链接时的状态

增加了一个伪类,即:hover。这 4 个伪类都只能用在超链接上,其使用方法如下:

```
a:visited{样式属性:属性值;样式属性:属性值;…}
a:active{样式属性:属性值;样式属性:属性值;…}
a:hover{样式属性:属性值;样式属性:属性值;…}
a:link{样式属性:属性值;样式属性:属性值;…}
```

说明:以上代码,在 visited、active、hover 和 link 之前都有一个冒号(:),这个冒号就是伪类和伪元素的标记符,所有的伪类和伪元素都是以冒号开头的。冒号前为该伪类或伪元素作用的元素的元素名。表 12-1 所示为 CSS 2 中的伪类。

表 12-1 CSS 2 中的伪类

伪 类	作 用
:link	设置超链接未被访问前的样式,如果超链接无 href 属性,则该样式不起作用
:hover	设置鼠标停放在该元素上时的样式,通常也用于超链接,但在 CSS 2 中该伪类还可以用在其他对象上。当作用在超链接上时,如果超链接无 href 属性,则该样式不起作用
:active	设置鼠标单击时的样式,即鼠标按下去还没有释放的那一瞬间的样式。通常也用于超链接,但在 CSS 2 中该伪类还可以用在其他对象上,当作用在超链接上时,如果超链接无 href 属性,则该样式不起作用
:visited	设置超链接已被访问的样式,如果超链接无 href 属性,则该样式不起作用
:focus	设置对象获得焦点时的样式
:first-child	设置某标签的第一个子标签的样式
:first	设置页面容器的第一页使用的样式,通常用于打印控制
:left	设置页面容器位于装订线左边的所有页面使用的样式,通常用于打印控制
:right	设置页面容器位于装订线右边的所有页面使用的样式,通常用于打印控制
:lang	设置对象使用特殊语言的内容的样式

2. 伪元素

伪元素和伪类的使用方法类似,伪元素对插入到文档中的虚构元素进行样式设置。

常用的伪元素有 first-letter 和:first-line,这两个伪元素可以将样式作用在文字的首字与首行,通常用在块级元素中。

在 CSS 2 中一共定义了 4 个伪元素,伪元素名及其作用如表 12-2 所示。

表 12-2 CSS 2 中的伪元素

伪 元 素	作 用
:first-letter	设置对象中的第一个字母的样式,仅用在块元素中
:first-line	设置对象中的第一个文字的样式,仅用在块元素中
:before	设置在对象前发生的内容,用来与 content 属性(CSS 2 中的属性)一起使用
:after	设置在对象后发生的内容,用来与 content 属性一起使用

12.1.4 CSS 的优先级

由于样式具有多种选择符,而选择符之间又有继承性与层叠性,在设计样式时,就有可将多个样式加载在同一个标签元素上,如果这些样式都不相同,则该标签元素可以同时拥有

这几种样式，但如果样式相同，只是属性值不同，就会产生样式冲突。

在 CSS 中对于每种不同类型的样式选择符都有一个特殊性(specificity)，特殊性使用相对权重(weight)(也就是优先级)来描述不同的样式选择符。

CSS 可以根据产生冲突的样式选择符的权重来判断使用哪种样式，通常是选择权重大的样式而忽略权重小的样式。在 CSS 2.1 中使用一个 4 位的数字串来表示权重，以下是有关权重的一些规定：

- 类型选择符(E)的权重为 0001。
- 类选择符(.classname)的权重为 0010。
- ID 选择符(#idname)的权重为 0100。
- 通用选择符(*)的权重为 0000。
- 子选择符的权重为 0000。
- 属性选择符([attr])的权重为 0010。
- 伪类选择符(:pseud-classes)的权重为 0010。
- 伪元素(:pseud-elements)的权重为 0001。
- 包含选择符的权重为包含的选择符权重值之和。
- 内联样式的权重为 1000。
- 继承的样式的权重为 0000。

以上权重由大至小依次是 1000、0100、0010、0001、0000。

12.1.5 CSS 中的单位

CSS 中的单位可以简单地分为颜色单位、长度单位、时间单位、角度单位和频率单位 5 种。

1. 颜色单位

在 CSS 中经常会用到颜色，而表达颜色的方式主要有#RRGGBB、rgb(R,G,B)和颜色名称。

#RRGGBB 表示方法是比较常用的一个表示方法，其中 RR 代表红色值、GG 代表绿色值、BB 代表蓝色值，取值范围都是 00～FF。例如，红色可以用"#FF0000"来表示。

rgb(R,G,B)是颜色的另一种表示方法，其中 R 代表红色值、G 代表绿色值、B 代表蓝色值，取值范围都是 0～255 或 0%～100%。例如，红色可以用 rgb(255,0,0)或 rgb(100%,0%,0%)来表示。使用百分数的表示方法不是所有浏览器都支持的。

使用颜色名称来表达颜色是比较直观的，例如红色可以直接用 red 来表示，但不同的浏览器会有不同的预定义的颜色名称。

2. 长度单位

在 CSS 中长度单位分为两种，一种是绝对长度单位，另一种是相对长度单位。绝对长度单位包括 pt、cm、mm、in 和 pc 等。

- in：英寸(inch)，常用的度量单位。
- pt：磅，这是标准的印刷量度，广泛使用在打印与排版上，72 磅相当于 1 英寸。

- cm：厘米(centimeter)，全世界统一的度量单位，1 英寸等于 2.54 厘米，1 厘米等于 0.394 英寸。
- mm：毫米(millimeter)，全世界统一的度量单位，1 厘米等于 10 毫米。
- pc：派卡(pica)，相当于我国新四号铅字大小。

以上 5 种绝对长度单位的换算方法为 1in = 2.54cm = 25.4mm = 72pt = 6pc = 12pc。

相对长度单位包括 px、ex 和 em 等。

- px：像素(pixel)，相对于显示器屏幕的分辨率而言。
- ex：相对于字符“x”的高度，该高度通常为字体尺寸的一半。
- em：相对于当前对象内文本的字体尺寸。

3. 时间单位

在 CSS 中时间单位只有两种，即 s(秒)和 ms(毫秒)，其中 1s=1000ms。

4. 角度单位

在 CSS 中角度单位包括 deg、grad 和 rad 等。

- deg：就是平常所说的“度”，一个圆等于 360deg。
- grad：梯度，一梯度为一个直角的百分之一，一个圆等于 400grad。
- rad：弧度，把一个圆分为 2PIrad。

5. 频率单位

在 CSS 中频率单位只有两种，即 Hz(赫兹)和 kHz(千赫兹)，它们都是声波单位，其中 1kHz=1000Hz。

12.2 文字和文本样式

在 CSS 样式中，最基本的属性是用来设置文字和文本的样式。“文字”指的是单个文字或单词，“文本”指的是由文字组成的内容。为字体设置样式主要是设置字、词的样式，为文本设置样式主要是对整段文章设置样式。

12.2.1 设置文字样式

在 CSS 中对文字样式的设置有文字字体、文字大小、文字粗体以及文字斜体等的设置。

1. 设置字体

在 HTML 中可以使用<font face="字体名">设置文字字体，而在 CSS 中设置字体所用的属性是 font-family，其语法如下：

```
font - family:"字体 1" "字体 2" "字体 3"...
```

说明：在这里，可以为文字设置多个字体，当在运行页面的浏览器中找不到第一种字体

的时候就会使用第二种字体显示；如果找不到第二种字体，则会以第三种字体显示，依此类推。如果设置的几种字体在浏览器中都无法找到，会自动以浏览器设置的默认字体显示。

2. 设置文字大小

设置文字大小是指为页面中的文字设置绝对大小或相对高度。

使用 CSS 样式设置文字大小的语法如下：

```
font - size:文字的大小
```

说明：此处的文字大小可以是相对高度，也可以是绝对大小，相对高度是指文字相对于父对象文字尺寸来设置，包括 larger 和 smaller，使用成比例的 em 单位计算；绝对高度设置的是固定的大小，包括 xx-small、x-small、larger 等。

除了使用英文单词表示文字大小之外，还有一种文字大小的设置方式，就是使用具体的长度值或百分比。表 12-3 显示了在 CSS 样式表中设置文字大小的属性值、含义以及规则。

表 12-3 CSS 样式表中可以设置的文字大小

类 型	font-size 取值或单位	表 示 的 含 义
用英文单词表示绝对大小	xx-small	极小
	x-small	很小
	small	小
	medium	中
	larger	大
	x-large	很大
	xx-large	极大
	larger	较大，一般比父对象中的字体大一些
	smaller	较小，一般比父对象中的字体小一些
采用具体的长度值(浮点数+单位)	pt	点，1 点=1/72 英寸
	px	像素
	in	英寸
采用百分比	%	相对于父对象中字体的尺寸的比例

当浏览器窗口设置的默认字体变大的时候，只有设置了固定像素值的文字大小是绝对不变的。也就是说，如果希望在页面中显示的文字不随浏览器的设置而变化，则需要使用具体的长度值来设置文字的大小。

3. 设置粗体

在页面中经常使用加粗的字体表示强调，但是在 HTML 标记中加粗的程度只有一种，通过 CSS 样式可以为文字设置不同程度的加粗效果，其语法如下：

```
font - weight:字体的粗度
```

说明：在这里，字体的粗度可以使用数值表示，也可以使用英文单词表示。其具体的取值及含义如表 12-4 所示。

表 12-4 字体粗细取值

字体的粗细取值	取值的含义
100～900	数值越小，字体越细，要求所取的数值是整百的，即 100、200、300 等
normal	正常字体效果
bold	加粗字体，字体的粗细与设置为 700 基本相同
bolder	特粗字体，就是在加粗字体的基础上再加粗，基本相当于设置为 900 的效果
lighter	细体字，相对默认字体更细一些

4. 设置文字颜色

在 CSS 样式表中设置文字颜色的属性是 color，其语法如下：

```
color:颜色代码/颜色名称
```

说明：这里的颜色代码是指颜色的十六进制数，颜色名称是颜色的英文名。

5. 设置斜体

在 CSS 样式表中也可以将文字设置为斜体显示，而且倾斜的程度有两种，即倾斜字体和偏斜体。其设置语法如下：

```
font - style:normal/italic/oblique
```

说明：font-style 可以取 normal（正常字体）、italic（倾斜）和 oblique（偏斜体）3 种值。

6. 综合应用

前面介绍的几种属性都是以 font 开始的，表示这几种属性属于同一类别，都是用来设置文字的字体效果的。在 CSS 样式表中还可以很方便地设置字体属性，即直接使用 font 属性进行设置，其语法如下：

```
font:字体属性取值
```

说明：在这里，字体属性取值可以直接设置各种属性值，各属性值之间用空格隔开。

12.2.2 设置文本样式

文本样式设置是对一段文字整体进行设置。文本样式设置包括设置阴影效果、大小写转换、文本缩进、文本对齐方式等。

1. 设置阴影效果

在 CSS 2 中允许设置文字的阴影，让文字看起来更有立体感。设置阴影所用的属性为 text-shadow，其语法如下：

```
text - shadow:none | color | length | inherit
```

说明：以上代码的属性值所代表的含义如下。

- none：不设置阴影。

- color：阴影的颜色。
- length：长度值。
- inherit：继承父级样式。

CSS 中的阴影有 3 个 length 需要进行设置，第 1 个是水平方向的距离，可以为负值；第 2 个是垂直方向的距离，可以为负值；第 3 个为模糊半径的长度，不能为负值。

2. 大小写转换

在 CSS 中处理大小写都是通过 text-transform 属性完成的，其语法如下：

```
text - transform:capitalize | uppercase | lowercase | none | inherit
```

说明：以上代码的属性值所代表的含义如下。

- capitalize：将每个文字的第一个字母大写。
- uppercase：将整个文字都变成大写。
- lowercase：将整个文字都变成小写。
- none：不改变文字的大小写。
- inherit：继承父级样式。

3. 文本缩进

在没有使用 CSS 之前，一段文字的首行缩进都是使用空格实现的；在有了 CSS 之后，网页开发者就不再需要在每个段落之前都加两个空格了；使用 CSS 中的 text-indent 属性可以轻易地达到缩进的目的。text-indent 属性的语法如下：

```
text - indent:length|百分数|inherit
```

说明：以上代码的属性值所代表的含义如下。

- length：缩进量，可以使用绝对单位值与相对单位值。
- 百分数：相对于父级元素的百分之多少来缩进。
- inherit：继承父级样式。

4. 文本的水平对齐方式

使用 text-align 属性可以在 CSS 样式表中设置文本的水平对齐属性，包括左对齐、右对齐、居中对齐和两端对齐，其设置语法如下：

```
text - align:left | right | center | justify
```

5. 文本的垂直对齐方式

文本的垂直对齐属性 vertical-align 相当于 HTML 中的垂直对齐标记，用于设置文本和其他元素（一般是上一级元素或者同行的其他元素）的垂直对齐方式，其语法如下：

```
vertical - align:baseline | sub | super | top | bottom | text - top | middle | text - bottom |百分比
```

说明：对属性值的详细说明见表 12-5。

表 12-5 垂直对齐的取值含义

纵向对齐的取值	具体含义
baseline	设置文本和上级元素的基线对齐
sub	设置文本显示为上级元素的下标，常在数组中使用
super	设置文本显示为上级元素的上标，常用于设置某个数值的乘方数
top	使文本元素和同行中最高的元素上端对齐
bottom	使文本元素和同行中高度最低的元素向下对齐
text-top	使文本元素和上级元素的文本向上对齐
middle	使文本垂直居中对齐，假如元素的基线与上级元素的 X 高度的一半相加的值为 H，则文本与高度 H 的中点纵向对齐。其中，X 指字母“X”的高度
text-bootom	使文本元素和上级元素的文本向下对齐
百分比	相对于元素行高属性的百分比，它会在上级元素基线上增加指定的百分比，如果取值为正数，则表示增加设置的百分比，如果取值为负数，则表示减少相应的百分比

6. 设置文本流入方向

CSS 中的 direction 属性可以用来设置文本流入的方向，direction 属性的语法如下：

```
direction:ltr | rtl | inherit
```

说明：以上代码的属性值所代表的含义如下。

- ltr：left to right 的简写，用于设置文本从左到右流入，该值为 direction 属性的默认值。
- rtl：right to left 的简写，用于设置文本从右到左流入。
- inherit：继承父级样式。

文本流入方向与水平对齐不同的是，句号总是放在文本流入方向的最后面，文本流入方向为从左向右流入时，句号在文本的右侧；文本流入方向为从右向左流入时，句号在文本的左侧。

7. 设置文本修饰

文本修饰一般包括设置文字带有下划线、上划线、删除线等，这些都可以使用 text-decoration 属性来设置，其语法如下：

```
text - decoration:underline | overline | line - through | blink | none
```

说明：text-decoration 属性可以取 5 种值，其含义分别见表 12-6。

表 12-6 文本的修饰属性

文本修饰属性值	取值的具体含义
underline	给文字添加下划线效果
overline	给文字添加上划线效果
line-through	给文字添加删除线效果
blink	给文字添加闪烁效果，只有在 Netscape 浏览器中才能看到效果
none	不设置任何修饰属性

12.2.3 空白与换行

在 HTML 代码中通常会出现很多空格与换行，这些空格与换行在浏览器中显示时往往不按照源代码中的出现方式来显示。在 CSS 中可以设置如何处理这些空格与换行。

1. 空格的处理方式

在 HTML 中，浏览器会自动将多个连续空格处理成一个空格，也可以使用 pre 元素让浏览器在显示时不更改源代码里的排版方式。这些在 CSS 中都可以统一使用 white-space 属性来完成，white-space 属性的语法如下：

```
white-space:normal | pre | nowrap | inherit
```

说明：以上代码的属性值所代表的含义如下。

- normal：默认值，浏览器会自动忽略多余的空格，连续多个空格只显示一个。
- pre：与 pre 元素类似，浏览器不忽略源代码中的空格。
- nowrap：设置文字不自动换行。
- inherit：继承父级样式。

2. 字内换行

当文本宽度超出浏览器宽度的时候，在默认情况下会自动换行，但如果正好是在较长的英文单词中间，那么整个单词都会被移动到下一行显示，这样本行的右侧就有了较大的空白，影响美观，使用字内换行属性可以将英文单词打散显示，也可以设置在换行前或换行后整体显示，其设置语法如下：

```
word-break:normal | break-all | keep-all
```

说明：normal 是正常情况下的显示方式，当在单词中需要换行的时候，该单词会在下一行显示，而本行后面保留空白；break-all 允许非亚洲语言文本行的任意字内断开；keep-all 与所有非亚洲语言的 normal 相同，对于中文、韩文、日文不允许断开。

12.2.4 设置间距

在 CSS 中可以定义文字与文字之间的距离，其中包括行间距、字间距与词间距，不同的间距可以控制页面的不同显示效果。

1. 行间距

行间距是指文本行与行之间的距离，在 CSS 中不能直接定义行间距，只能通过 line-height 属性定义行高。所谓行高是指上一行文字的基线与下一行文字的基线之间的距离，行高等于行间距加上文字高度。line-height 属性的语法如下。

```
line-height:normal | number | length |百分数| inherit
```

说明：以上代码的属性值所代表的含义如下。

- normal：默认值，使用的是默认行高。
- number：在当前文字大小的基础上做增加来设置行高，不能为负值。
- length：指定行高数，可以是绝对长度单位，也可以是相对长度单位，不能是负值。
- 百分数：用百分数指定行高，相当于字体大小的百分之多少。
- inherit：继承父级属性。

2. 字间距

在 CSS 中可以通过 letter-spacing 属性来设置字间距。对于英文来说，字间距是指每个字母之间的距离，对于中文来说，字间距是每个字之间的距离。letter-spacing 属性的语法如下：

```
letter - spacing:normal | length | inherit
```

说明：以上代码的属性值所代表的含义如下。

- normal：默认值，使用默认的字间距。
- length：设置字间距，可以是绝对单位也可以是相对单位。
- inherit：继承父级属性。

3. 词间距

在 CSS 中可以使用 word-spacing 设置词间距，词间距是针对英文而言的。目前，浏览器还不能区分中文的“词”与“字”，word-spacing 属性的语法如下：

```
word - spacing:normal | <length>| inherit
```

说明：以上代码的属性值所代表的含义如下。

- normal：默认值，使用默认的词间距。
- length：设置词间距的大小，可以是绝对单位值也可以是相对单位值。
- inherit：继承父级属性。

12.2.5 CSS 注释

CSS 中的注释与 HTML 中的注释有所不同，CSS 中的注释采用的格式如下：

```
/ * 注释内容 * /
```

说明：注释可以是单独的一行，也可以跨行，但不能嵌套。

12.3 设置表格、列表和滚动条样式

表格、列表和滚动条是网页设计中比较常用的一些元素。在 CSS 中提供了许多属性进行表格、列表和滚动条样式的设置，通过设置这些样式可以让网页内容更加吸引浏览者的注意。

12.3.1 设置表格样式

在 CSS 中有一些样式是在表格里使用得比较多的，在此统称为表格样式。这些样式可以实现合并边框、设置边框间距、设置表格标题的位置、设置表格布局等功能。

1. 边框

在一个表格中同时存在着两种边框，一种是表格的表框，即表格最外面的 4 条边框；另一种是单元格的边框，每一个单元格都有自己的边框。在默认情况下，这两种不同的边框是分开显示的，但在 CSS 的 border-collapse 属性的作用下，可以将这两种边框合并起来。border-collapse 属性的语法如下：

```
border-collapse:collapse | separate | inherit
```

说明：以上代码的属性值所代表的含义如下。

- collapse：合并两种边框。
- separate：两种边框独立，该值为默认值。
- inherit：继承父级样式。

2. 定义表格边框的间距

在 CSS 中可以使用 border-spacing 属性为表格设置边框间距，这一点与 HTML 中 table 元素的 cellspacing 属性十分类似。border-spacing 属性的语法如下：

```
border-spacing:宽度|inherit
```

说明：以上代码的属性值所代表的含义如下。

- 宽度：边框间距的大小，可以是绝对单位值也可以是相对单位值，但不能是负数。
- inherit：继承父级样式。

只有当 border-collapse 属性值为 separate，或没有设置 border-collapse 属性值时，border-spacing 属性才会生效，否则该属性不会产生作用。IE 浏览器不支持 border-spacing 属性。

3. 定义表格标题的位置

在 HTML 中可以使用 caption 元素设置表格的标题，而在 CSS 中 caption-side 属性可以用来设置将标题放在表格的什么位置，caption-side 属性的语法如下：

```
caption:top | bottom | left | right | inherit
```

说明：以上代码的属性值所代表的含义如下。

- top：标题位于表格顶部。
- bottom：标题位于表格底部。
- left：标题位于表格左侧。
- right：标题位于表格右侧。
- inherit：继承父级样式。

使用 caption 元素的 align 属性和 valign 属性同样可以将表格标题放在表格的不同位置，不同的浏览器对这两个属性的支持情况不完全一样。

4. 设置表格布局

当一个单元格里对象的宽度超过单元格所定义的宽度时，在能换行时(如文字)，浏览器会自动在宽度的最大处换行，在不能换行时(如图片或一个超长单词)，浏览器会自动调整表格列的宽度，以容纳单元格中的对象。在 CSS 中有一个名为 table-layout 的属性可以设置是否保证单元格宽度不被改变，table-layout 属性的语法如下：

```
table-layout:auto | fixed | inherit
```

说明：以上代码的属性值所代表的含义如下。

- auto：当内容超过宽度时，如能自动换行则自动换行，如不能自动换行则增加宽度，该值为默认值。
- fixed：无论内容是否超过宽度，都保持原来的宽度。
- inherit：继承父级样式。

12.3.2 设置列表样式

在 CSS 中有专门的设计列表的样式，使用这些样式可以用图片代替列表前的标号，也可以用不同的方式显示列表前的标号，还可以设置列表文字的排列方式以及间距。

1. 设置列表符号样式

在 HTML 中的列表符号只能是一个黑点或数字，显得十分单调，使用 CSS 中的 list-style-type 属性可以指定符号的样式，其语法如下：

```
list-style-type:circle | disc | decimal | square | upper-roman | lower-roman | upper-
alpha | lower-alpha | none | armenian | cjk-ideographic | georgian | hebrew | lower-greek |
hiragana | hiragana-iroha | katakana | katakana-iroha | lower-latin | upper-latin
```

说明：以上代码的属性值所代表的含义如下。

- circle：显示空心圆标号。
- disc：默认值，显示实心圆标号。
- decimal：显示阿拉伯数字。
- square：显示实心方块标号。
- upper-roman：显示大写罗马数字。
- lower-roman：显示小写罗马数字。
- upper-alpha：显示大写英文字母。
- lower-alpha：显示小写英文字母。
- none：不使用项目符号。

由于后面的属性目前的主流浏览器都不支持，所以此处不再说明。

2. 使用图片设置列表样式

除了可以采用系统提供的一些列表符号，在 CSS 中还可以利用 list-style-image 属性设置图像作为列表符号，其语法如下：

```
list - style - image:url(源文件地址)
```

说明：此处属性值为图片的地址，为了使列表符号能够清晰显示，不要选择过大的图片，图片格式为 JPEG、GIF、PNG 几种。

3. 列表符号的显示位置

当在列表中使用了文本样式（如背景颜色等）时，可以用 list-style-position 属性指定符号的显示位置，即指定符号是放在文本块之外还是放在文本块之内，list-style-position 属性的语法如下：

```
list - style - position:outside | inside
```

说明：以上代码的属性值所代表的含义如下。

- outside：将列表符号放在文本块之外，该值为默认值。
- inside：将列表符号放在文本块之内。

list-style-position 属性可以作用在<ol>、<ul>和<li>标签上。

4. 综合设置列表样式

在 CSS 中可以使用 list-style 属性综合设置列表的所有样式，使用 list-style 为列表设置样式可以不输入 list-style-image、list-style-type 或 list-style-position 属性名，直接输入属性值来简化输入即可。list-style 属性的语法如下：

```
list - style:list - style - image | list - style - type | list - style - position
```

说明：在使用 list-style 设置列表样式时要注意以下两点。

- 当同时指定 list-style-image 和 list-style-type 时，list-style-image 将优先显示，除非 list-style-image 为 none，或图片地址错误而无法显示。
- 当列表与列表项同时使用样式时，列表项的样式将优先显示。

与 list-style-image、list-style-type 和 list-style-position 样式相同，list-style-position 样式可以作用在<ol>、<ul>和<li>标签上。

12.3.3 设置滚动条样式

滚动条一般都有立体效果，这个效果是通过边框的亮暗对比来体现的，亮的边框就好像是光照到的地方，暗的边框就好像是由于光线被遮挡出现的阴影效果，利用 CSS 中的滚动条属性可以设置滚动条的各种颜色效果，例如滚动条的边框颜色、表面效果等。

1. 设置滚动条颜色

使用 scrollbar-face-color 属性可以设置滚动条的颜色，其语法如下：

```
scrol lbar - face - color:颜色
```

说明：scrollbar-face-color 的属性值只有一个，就是颜色，其值可以是十六进制的 RGB 颜色、颜色的英文名或用百分比表示颜色。

2. 设置滚动条亮边框颜色

使用 scrollbar-highlight-color 属性可以设置滚动条亮边框的颜色，也就是滚动条左边和上边边框的颜色，其语法如下：

```
scrol lbar - highlight - color:颜色
```

说明：scrollbar-highlight-color 的属性值只有一个，就是颜色，其值可以是十六进制的 RGB 颜色、颜色的英文名或用百分比表示颜色。

3. 设置滚动条暗边框颜色

用户可以设置亮边框颜色就可以设置暗边框颜色，也就是滚动条右边和下边边框的颜色。如果要设置暗边框的颜色，可以使用 scrollbar-shadow-color 属性，scrollbar-shadow-color 属性的语法如下：

```
scrol lbar - shadow - color:颜色
```

说明：scrollbar-shadow-color 的属性值只有一个，就是颜色，其值可以是十六进制的 RGB 颜色、颜色的英文名或用百分比表示颜色。

4. 设置滚动条方向箭头颜色

scrollbar-arrow-color 属性可以用来设置滚动条的方向箭头的颜色，其语法如下：

```
scrol lbar - arrow - color:颜色
```

说明：scrollbar-arrow-color 的属性值只有一个，就是颜色，其值可以是十六进制的 RGB 颜色、颜色的英文名或用百分比表示颜色。

5. 设置滚动条基准颜色

scrollbar-base-color 属性可以用来设置滚动条的基准颜色，在设置了基准颜色之后，滚动条的其他颜色都会根据该颜色自动调整，包括箭头颜色、边框颜色等，其语法如下：

```
scrol lbar - base - color:颜色
```

说明：scrollbar-base-color 的属性值可以是十六进制的 RGB 颜色、颜色的英文名。

12.4 设置背景、边框、边距和补白

背景颜色、背景图片、边框和边距是网页设计中用得比较多的修饰方法，合理地配置网页的前景色与背景色，再加上边框和边距的辅助，可以让网页看起来更漂亮。

12.4.1 设置背景

1. 背景颜色

背景通常指除了文本与边框之外的所有颜色。在 CSS 中可以使用 background-color 来设置背景颜色，background-color 属性的语法如下：

```
background - color:transparent |颜色| inherit
```

说明：以上代码的属性值所代表的含义如下。

- transparent：设置背景颜色透明，该值为默认值。
- 颜色：可以为命名颜色、RGB 颜色或百分比颜色。
- inherit：继承父级样式。

在 HTML 中大多数元素都可以设置背景颜色，例如 body、div、td 等，transparent 属性用于设置背景透明，也可以理解为没有背景颜色。

2. 设置背景图像

在 HTML 中设置网页背景图片的方式为<body background="图片 URL">，在 CSS 中设置背景图片的属性为 background-image，该属性不仅可以设置网页背景图片，还可以设置表格、单元格、按钮等元素的背景图片。background-image 属性的语法如下：

```
background - image:none | url(url) | inherit
```

说明：以上代码的属性值所代表的含义如下。

- none：无背景图片，该值也是默认值。
- url(uri)：图片的 url 地址，可以是绝对地址也可以是相对地址。
- inherit：继承父级样式。

3. 设置固定背景图像

通常，在网页上设置了背景图片之后，背景图片都会平铺在网页的下方，若网页内容比较多，在拖动滚动条时，网页的背景会跟着网页的内容一起滚动。在 CSS 中使用 background-attachment 属性将背景图片固定在浏览器上，此时如果拖动滚动条，背景图片不会随着网页内容滚动而滚动，看起来好像文字浮动在图片上似的。background-attachment 属性的语法如下：

```
ground - attachment:scroll | fixed | inherit
```

说明：以上代码的属性值所代表的含义如下。

- scroll：背景图片随着内容滚动，该值为默认值。
- fixed：背景图片固定，不随着内容滚动。
- inherit：继承父级样式。

4. 设置背景图像的平铺方式

在 HTML 中，如果背景图片大小小于浏览器窗口大小，浏览器会自动将背景图片平

铺，以充满整个浏览器窗口。不过在很多情况下，这种方式并不是最好的背景图片展现方式。在 CSS 中可以通过 background-repeat 属性来设置背景图片的平铺方式，background-repeat 属性的语法如下：

```
background - repeat:repeat| no - repeat | repeat - x | repeat - y | inherit
```

说明：以上代码的属性值所代表的含义如下。

- repeat：平铺背景图片，该值为默认值。
- no-repeat：不平铺背景图片。
- repeat-x：背景图片在水平方向上平铺。
- repeat-y：背景图片在垂直方向上平铺。
- inherit：继承父级样式。

5. 背景图像的定位

在默认情况下，背景图像都是从元素的左上角开始显示的，使用 background-position 属性可以更改背景图像开始显示的位置，其语法如下：

```
background - position:位置的具体值
```

在这里设置图像位置的属性值有多种形式，可以是 X、Y 轴方向的百分比或绝对值，也可以使用表示位置的英文名称，具体取值及其含义见表 12-7。

表 12-7 设置背景图像的位置

取值方式	具 体 含 义
百分比(X%Y%)	起始位置与左上角的距离占整个元素的比例，包括水平方向和垂直方向。例如设置网页的背景图像，则会以整个页面的大小为依据
绝对数值(x,y)	起始位置的绝对坐标，包括横坐标和纵坐标，这是以左上角为端点的，在使用这种格式的时候需要同时设置长度单位
top	使图像在垂直方向上居于顶端
bottom	使图像在垂直方向上居于底部
left	使图像在水平方向上居于左端
right	使图像在水平方向上居于右端
center	图像在中部显示，它可以设置为水平方向，也可以设置为垂直方向

在这些设置方式中，百分比和绝对数值可以混用，即前面是百分比，后面可以是数值；同样前面是数值，后面可以是百分比。

12.4.2 设置边框

表格的边框很容易理解，其实在 HTML 中很多对象都是有边框的，例如 div、input 等。在 HTML 中，这些元素的边框都是很呆板的，甚至有些元素显示不了边框，在有了 CSS 之后，网页开发者就可以很轻松地设置边框的样式了，例如边框的粗细、颜色等。

1. 设置边框样式

边框的样式在边框的几个属性里可以说是最重要的，边框样式除了可以改变 HTML

中呆板的边框样式之外，在某些时候甚至可以控制边框是否显示。在 CSS 中设置边框样式的属性为 border-style，该属性的语法如下：

```
border-style:边框的样式值
```

说明：边框样式的具体取值见表 12-8。

表 12-8 边框的样式值

属 性 值	具 体 含 义
none	无边框
solid	实线的效果
dotted	点线效果，即边框由点组成
dashed	划线效果，即边框由多个短线组成
double	双实线效果
groove	带立体效果的沟槽
ridge	突出的脊形效果
inset	内嵌一个立体的边框
outset	外嵌一个立体的边框

2. 设置不同的边框样式

使用 border-style 属性也可以为对象的 4 个边框设置不同的样式，其设置方法与 border-width 属性类似，可以直接使用 border-style 属性设置 4 个边框的风格，它们对应的边框依次是上边框、右边框、下边框和左边框，如果这里只设置了一个边框风格，则会对 4 个边框同时起作用；如果设置了两个，则第 1 个值应用于上、下边框，第 2 个值应用于左、右边框；如果设置 3 个，第 1 个用于上边框，第 2 个用于左、右边框，第 3 个用于下边框。

3. 设置边框宽度

在 HTML 中 table 元素可以使用 border 属性来设置边框的宽度，在 CSS 中可以使用 border-width 属性来设置边框宽度，但是 border-width 属性不仅可以设置表格的边框宽度，还可以设置任何一个有边框的对象的边框宽度，border-width 属性的语法如下：

```
border-width:medium | thin | thick | 数值
```

说明：以上代码的属性值所代表的含义如下。

- medium：默认宽度，该值为默认值。
- thin：比默认宽度小。
- thick：比默认宽度大。
- 数值：以绝对单位数值或相对单位数值来指定边框的宽度。

4. 设置不同的边框宽度

使用 border-width 属性不仅可以设置整个边框宽度，还可以设置单个边框的宽度，用法和设置边框样式一样，如果这里只设置了一个边框宽度，则会对 4 个边框同时起作用；如

果设置了两个边框宽度，则第 1 个值应用于上、下边框，第 2 个值应用于左、右边框；如果提供 3 个边框宽度，第 1 个用于上边框，第 2 个用于左、右边框，第 3 个用于下边框。

5. 设置边框的颜色

border-color 属性不仅可以为表格设置边框颜色，还可以为几乎所有的块对象添加边框颜色，例如 p、div 等元素，border-color 属性的语法如下：

```
border-color:颜色| transparent
```

说明：以上代码的属性值所代表的含义如下。

- 颜色：边框的颜色，可以是颜色英文名、RGB 颜色或用百分比表示颜色。
- transparent：透明颜色，即不设置颜色。

6. 设置不同的边框颜色

使用 border-color 属性不仅可以统一设置 4 个边框的颜色，还可以设置单个边框的颜色，其设置方法与 border-width 属性和 border-style 属性类似。

7. 综合设置边框效果

在 CSS 中还可以使用 border 属性直接设置边框的整体效果，其语法如下：

```
border:边框宽度  边框样式  边框颜色
```

在这里，可以只设置其中的一项或几项，但如果要正常显示设置的边框效果，需要设置边框的样式，即使是采用默认的 solid。

12.4.3 设置边距

边距和补白都是为控制页面的松紧程度而提供的属性，边距一般是设置元素周围的边界宽度，这个宽度可以明显地区分不同的元素，也可以让网页中的内容没有那么拥挤。

1. 设置上边距

在 CSS 中可以为一个元素设置各个方向的边界宽度。上边距指元素与它上面的元素之间的距离，采用的是 margin-top 属性，其设置语法如下：

```
margin-top:距离值
```

说明：这里的距离值可以是百分比，也可以是由数值和单位组成的确定的距离，如果只给出一个数值，则默认其单位是像素，百分比是以该元素的上一级元素为基础进行设置的。

2. 设置下边距

下边距和上边距相对，指元素距离下方元素的边距值，其语法如下：

```
margin-bottom:距离值
```

说明：这里的距离值可以是百分比，也可以是由数值和单位组成的确定的距离，如果只

给出一个数值，则默认其单位是像素。

3. 设置左边距

左边距就是元素距离左侧其他元素的距离，其语法如下：

```
margin-left:距离值
```

说明：这里的距离值可以是百分比，也可以是由数值和单位组成的确定的距离，如果只给出一个数值，则默认其单位是像素。

4. 设置右边距

右边距就是元素距离右侧其他元素的距离，其语法如下：

```
margin-right:距离值
```

说明：这里的距离值可以是百分比，也可以是由数值和单位组成的确定的距离，如果只给出一个数值，则默认其单位是像素。

5. 综合设置边距

如果要同时设置某个元素的4个边距，除了可以分别进行设置外，还可以使用复合属性margin进行设置，其语法如下：

```
margin:各个边距的值
```

说明：在这里可以设置1～4个边距。如果设置一个值，则同时作用于元素的4个方向；如果设置两个值，则分别作用于上下和左右边距；如果设置3个值，则分别作用于上边距、左右边距和下边距；如果设置4个值，则按照上、右、下、左的顺序起作用。

12.4.4 设置补白

补白用于设置元素边框和内容之间的距离，是设置元素自身松紧程度的属性，可以理解为在盒子里增加填充物，以避免里面的东西被打破。

1. 设置顶端补白

顶端补白指元素的内容与其上边框的距离，常用来设置页面补白，其语法如下：

```
padding-top:距离值
```

说明：这里的距离值一般采用数值，可以为其添加单位；如果没有设置单位，默认以像素为单位。

2. 设置底部补白

底部补白就是设置页面元素距离下边框的距离，其语法如下：

```
padding-bottom:距离值
```

说明：这里的距离值一般采用数值，可以为其添加单位；如果没有设置单位，默认以像素为单位。

3. 设置左侧补白

设置左侧补白指设置页面中元素与左侧边界之间的间隔，其语法如下：

```
padding-left:距离值
```

说明：这里的距离值一般采用数值，可以为其添加单位；如果没有设置单位，默认以像素为单位。

4. 设置右侧补白

设置右侧补白指设置页面中元素与右侧边界之间的间隔，其语法如下：

```
padding-right:距离值
```

说明：这里的距离值一般采用数值，可以为其添加单位；如果没有设置单位，默认以像素为单位。

5. 综合设置补白

如果要同时设置某个元素的4个补白，除了可以分别进行设置外，还可以使用复合属性padding进行设置，其语法如下：

```
padding:各个方向的补白
```

说明：在这里可以设置1～4个补白值。如果设置一个值，则同时作用于4个方向；如果设置两个值，则分别作用于上下和左右方向；如果设置3个值，则分别作用于顶端补白、左右补白和底部补白；如果设置4个值，则按照上、右、下、左的顺序起作用。

12.5　控制元素布局

CSS可以用来控制元素的布局，通过对DIV的位置进行设置来定位网页元素。

12.5.1　块级元素和内联元素

1. 块级元素和内联元素的概念

块级元素生成的是一个矩形框，并且和相邻的块级元素依次竖直排列，不会排在同一行。例如<p>元素、<ul>元素、<hi>元素、<form>元素等都是块级元素，它们总是以一个块出现，总是单独占据一行。

内联元素通俗来说就是文本的显示方式，读者常用到的<a>、<img>、<input>都属于内联元素，内联元素的显示特点就是像文本一样显示，各个元素之间横向排列，到最右端自动换行，不会独自占据一行。当然，块级元素也能变成内联元素，这就要用到下面所讲的定位和浮动了。

2. <div>元素和<span>元素

为了更好地理解块级元素和内联元素，这里重点介绍CSS布局中经常使用的<div>元素和<span>元素，利用这两个元素，加上CSS对其样式的设计，可以很方便地实现各种效果。

1）<div>元素

<div>元素简单而言就是一个独立的对象，它是一个标准的块级元素，用它可以容纳各种元素，从而方便排版，在用CSS设置样式时只需要对<div>进行相应的控制，其中包含的各个元素都会随之改变。<div>元素的语法如下：

```
<div>
各种元素或文字
</div>
```

2）元素

元素和<div>元素一样，作为容器标记被广泛应用在HTML语言中，在和之间同样可以容纳各种HTML元素，从而形成独立的对象。<span>元素和<div>元素的区别在于<div>元素是一个块级元素，它包围的元素会自动换行；而<span>元素是一个内联元素，它包围的元素不会自动换行，<span>元素没有结构上的意义，纯粹是为了应用样式，当其他内联元素都不合适时，就可以使用<span>元素。<span>元素的语法如下：

```
<span>
各种元素或文字
</span>
```

12.5.2 定位

CSS中的网页布局使用的都是块形式，而块出现在网页中的哪个位置所采用的是定位的模式，定位(positioning)就是允许网页开发者精确定义元素出现的相对位置，这个相对位置可以是相对父级元素、另一个元素或浏览器窗口。

1. 定位模式

在CSS中可以使用position属性设置定位的模式，position属性的语法如下：

```
position:static | relative | absolute | inherit
```

说明：以上代码的属性值所代表的含义如下。

- static：静态定位模式，即无特殊定位。块以普通方式生成，块级元素生成的是一个矩形框，是文档流中的一个部分，而内联级框是由一个或多个行框的上下文生成的。这些行框流动于父级元素中，该值为默认值。
- relative：一个相对定位模式，使用该模式的块可以偏移一定的距离，块偏移的方向和幅度可以由top、left、right和bottom几个偏移属性联合指定，其产生过程是先用

static 方式生成一个块，再移动这个块到指定的相对位置。

- absolute：这是一个绝对定位模式，同样也是使用 top、left、right 和 bottom 几个属性来决定块的位置。
- inherit：继承父级样式。

2. 偏移

在定位模式中，relative、absolute 和 static 都需要使用偏移属性来指定定位的位置。在 CSS 中，偏移量有 4 个属性，即 left、right、top 和 bottom，分别代表左偏移量、右偏移量、上偏移量和下偏移量。其语法如下：

```
left:长度|百分比| auto | inherit
right:长度|百分比| auto | inherit
top:长度|百分比| auto | inherit
bottom:长度|百分比| auto | inherit
```

说明：以上代码的属性值所代表的含义如下。

- 长度：可以是绝对单位数值，也可以是相对单位数值，用于指明偏移的幅度。
- 百分比：以百分比的形式指定偏移幅度，这个百分比为父级元素的宽度和高度的百分比。
- auto：无特定的偏移量，由浏览器自己分配，该值为默认值。
- inherit：继承父级样式。

当一个元素被设置了偏移之后，这个元素的所有部分都会跟着一起偏移，如边框、边距、填充等，偏移量不仅可以为正值，还可以为负值。

3. 综合运用

在学习了定位和偏移之后，大家知道了定位有 4 种不同的模式，下面结合偏移分别介绍这几种模式的不同之处。

1）静态定位

静态定位模式是定位模式中的默认定位模式。在该模式中对定位没有任何要求，完全是由浏览器自动生成。对于块级元素来说，通常是生成一个矩形框，如 div 层等；对于内联元素来说，则按正常的流生成，如 b 元素等。

将元素的 position 属性值设为 static 可以设置元素的静态定位，由于静态定位模式并没有对元素在定位方面指出任何要求，因此所有的偏移属性在该模式下都是不起作用的。

2）绝对定位

绝对定位是相对于父级元素的 4 个边框而言的，通常可以把整个网页（或者说是 body 元素）看成一张纸，而绝对定位就是将块放在网页的某个位置。至于具体将块放在网页的哪个位置，由偏移量决定，将元素的 position 属性值设为 absolute 就可以设置元素的绝对定位。

3）相对定位

如果说绝对定位是相对网页的定位，那么相对定位就是相对元素自己的定位。所谓相对元素自己的定位是指元素相对于没有设置 position 属性之前的位置。将元素的 position

属性值设为 relative 可以设置元素的相对定位。

4. 定位元素的层叠次序

当一个页面内有多个层的时候就需要设置这些层的层叠顺序，这样才不会将页面中需要显示的内容遮挡住。在一般情况下，越晚添加的层，位置越靠上。设置层叠顺序的语法如下：

```
z-index:顺序号
```

在这里，层叠顺序是通过设置其所在的层顺序号来实现的。在一般情况下，取值为 1 表示该层位于最上层，也就是没有其他层覆盖该层。通常，顺序号越大，层越靠下，被覆盖的几率也就越大。

12.5.3 浮动

在一个网页文档里，文档流通常是从上到下、由左而右流动的。对于内联元素而言，在创建了一个元素之后会在其右接着创建其他元素；对于块级元素而言，在创建了一个元素之后会在其下方接着创建其他元素，CSS 中的浮动可以让某些元素脱离这种文档流的方式。

1. 浮动的概念

读者对于浮动的概念应该不会太陌生，在介绍图片和表格时曾介绍过图片和表格的对齐方式，这种对齐方式其实就是“浮动”。例如，“<img src="a.jpg" align="right">”会让图片向右方浮动，并且其他元素都会围绕着图片“流动”。在 HTML 中只有图片和表格可以浮动，而使用 CSS 可以让所有元素都浮动起来。

2. 设置浮动

在 CSS 中使元素浮动的属性为 float，其语法如下：

```
float:left | right | none
```

说明：以上代码的属性值所代表的含义如下。

- left：对象居左浮动，文本流向对象的右侧。
- right：对象居右浮动，文本流向对象的左侧。
- none：对象不浮动，该值为默认值。

3. 清除浮动

在一个元素被设为浮动之后，如果没有特别要求，这个元素之后的所有内容都会围绕该元素流动，此时需要清除图片的浮动。在 CSS 中可以使用 clear 属性清除浮动效果，其语法如下：

```
clear:none | left | right| both
```

说明：以上代码的属性值所代表的含义如下。

- none：不清除浮动，该值为默认值。

- left：不允许左边有浮动的元素。
- right：不允许右边有浮动的元素。
- both：左、右两侧都不允许有浮动的元素。

12.5.4 溢出与剪切

当一个元素的大小无法容纳其中的内容时就会产生溢出的情况，也就是元素中的内容已经显示在元素外面了，而剪切的作用是只显示元素中的某一部分，把其余部分都剪切掉。

1. 溢出内容的设置

在 CSS 中可以通过 overflow 属性处理溢出情况。overflow 属性的语法如下：

```
overflow:visible | hidden | scroll | auto | inherit
```

说明：以上代码的属性所代表的含义如下。

- visible：不剪切溢出的内容，也不添加滚动条。对于 IE 浏览器来说，会自动调整对象大小，以容纳其中内容，对于其他三大浏览器来说，会让溢出的内容显示在对象之外。该值为默认值。
- hidden：隐藏溢出的内容，用户将看不到溢出部分的内容。
- scroll：添加横向与纵向滚动条，用户可以通过拖动滚动条来查看溢出部分的内容。
- auto：浏览器决定使用哪个方法处理溢出的内容，通常在必要的时候显示滚动条。

2. 设置水平方向超出范围时的处理方式

使用 overflow 属性可以设置超出范围时的内容处理方式，但是一旦设置了，则对水平方向和垂直方向同时起作用，如果只需要设置其中一个方向，可以单独进行设置。使用 overflow-x 可以设置水平方向上的处理方式，其语法如下：

```
overflow - x:visible | auto | hidden | scroll
```

说明：

- visible：表示可见，即使内容超出了范围依然完整显示。
- auto：表示自动根据情况显示滚动条。
- hidden：表示裁切超出范围的内容。
- scroll：表示显示滚动条。

3. 设置垂直方向超出范围时的处理方式

使用 overflow-y 可以设置当内容超出元素的范围时在垂直方向上的处理方式，其设置语法如下：

```
overflow - y:visible | auto | hidden | scroll
```

说明：

- visible：表示可见，即使内容超出了范围依然完整显示。
- auto：表示自动根据情况显示滚动条。

- hidden：表示裁切超出范围的内容。
- scroll：表示显示滚动条。

4. 内容的剪切

在 CSS 中可以使用 clip 属性剪切对象，所谓“剪切”只是在对象上划分一个矩形的区域，属于该区域中的部分显示出来，不属于该区域的部分则隐藏。clip 属性的语法如下：

```
clip 属性:auto | rect(上 右 下 左) | inherit
```

说明：以上代码的属性值所代表的含义如下。

- auto：不剪切，该值为默认值。
- rect：根据上、右、下、左的次序划分一个区域，属于该区域内的部分显示，不属于该区域内的部分则隐藏，rect 的 4 个参数分别代表上、右、下、左 4 个边距。需要注意的是，这 4 个边距并不是指与上边框、右边框、下边框、左边框之间的距离，而是相对该对象的左上角坐标而言的距离。

12.5.5 对象的显示与隐藏

对于形状对象而言，除了可以设置溢出与剪切之外，还可以对整个块设置显示或隐藏。显示、隐藏与溢出、剪切不同，溢出与剪切所影响的只是对象的局部(当然也可以将局部扩大到全部)，而显示与隐藏影响的是整个对象。在 CSS 中可以使用 visibility(可见性)设置对象是否可见。visibility 属性的语法如下：

```
visibility:visible | hidden
```

说明：以上代码的属性值所代表的含义如下。

- visible：对象为可见的。
- hidden：对象为不可见的。

12.6 综合应用

12.6.1 登录界面

利用 HTML+CSS 制作管理系统的用户登录界面，使用 CSS 文档布局并设置样式。

(1) CSS 文档 loginstyle.css：

```
body
{font-size: small;
    background-image: url('bg.jpg');
    margin: auto;
    background-repeat: repeat-x;
    text-align: center;
}
#main
```

```
{margin: auto;
     width: 900px;
     text - align: center;
}
#Logo
{margin: auto;
     background - image: url('task_banner.gif');
     background - repeat: no - repeat;
     height: 110px;
     width: 884px;
}
#LoginMain
{margin: auto;
     background - image: url('userLoginBg.jpg');
     background - repeat: no - repeat;
     height: 300px;
     position: relative;
     top: 0px;
     left: 0px;
}
#Login_Table
{position: absolute;
     top: 120px;
     left: 219px;
     height: 77px;
     width: 394px;
     color: #FFFFFF;
     bottom: 103px;
}
.btn
{background - image:url('btn.jpg');
 height:72px;
 width: 90px;
}
```

(2) 登录页面的 HTML 代码(login.html):

```
<! DOCTYPE html PUBLIC " - //W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/
xhtml1/DTD/xhtml1 - transitional.dtd">
<html xmlns = "http://www.w3.org/1999/xhtml">

<head runat = "server">
    <title>登录界面</title>
    <link href = "loginstyle.css" rel = "stylesheet" type = "text/css" />
</head>
<body><div id = "main"><div id = "Logo"></div><div id = "LoginMain">
      <form id = "form1" runat = "server">
       <div id = "Login_Table">
       <table>
          <tr>
            <td>用户名:</td>
```

```
        <td><input type="text" id="usename" /></td>
        <td rowspan="3" colspan="3">
            <input type="button" name="sm" class="btn" />
        </td>
      </tr>
        <tr>
            <td>密码:</td>
            <td><input type="password" id="password" /></td>
        </tr>
        <tr>
          <td>验证码:</td>
          <td><input type="text" id="invcode" /></td>
         </tr>
    </table>
   </div>
   </form>
  </div></div></body></html>
```

(3) 代码在浏览器中运行的效果如图 12-1 所示。



图 12-1　用户登录界面的运行效果

12.6.2　花店 Banner

利用 HTML+CSS 制作花店的 Banner,使用 CSS 文档布局并设置样式。

(1) CSS 文档 css.css:

```
body{font-family:"微软雅黑"; font-size:14px; color:#F34;}

body,p,ul,li,h4,img{margin:0; padding:0; border:0; list-style:none;}
.banner{
width:1000px;
height:285px;
margin:13px auto 15px auto;
```

```
overflow:hidden;
}
.left{
width:755px;
height:285px;
font-weight:bold;
background:url(pic.gif);
position:relative;
float:left;
}
.content_left{
position:absolute;
top:90px;
right:45px;
text-align:right;
}
.school_en{
font-size:14px;
}
.school_ch{
font-size:24px;
font-family:"黑体";
padding-right:10px;
}
.advertise{
margin-top:20px;
font-family:"黑体";
font-size:16px;
}
ul.style_a{
margin-top:25px;
margin-left:120px;
list-style:none;
overflow:hidden;
}
ul.style_a li{
float:left;
margin-left:10px;
}
ul.style_a li a{
background:#FFF;
border:1px solid #ff7202;
width:26px;
height:22px;
text-align:center;
line-height:22px;
vertical-align:middle;
display:block;
color:#ff7202;
font-size:18px;
text-decoration:none;
```

```
}
ul.style_a li.current a{
width:30px;
height:26px;
line-height:26px;
background:#ff7202;
color:#FFF;
margin-top:-2px;
position:relative;
}
.right{
width:245px;
height:285px;
background:#fef0df;
float:right;
position:relative;
}
.aa {
font-size: 16px;
}
.bb {
color: #000;
text-decoration: line-through;
}

.content_right{
position: absolute;
top: 50px;
left: 30px;
height: 141px;
}
ul.style_icon{
margin-top:10px;
}
ul.style_icon li{
float:left;
margin-right:12px;
}
.cl{
clear:both;
margin-top:55px;
margin-right:30px;
text-indent:2em;
line-height:24px;
}
```

(2) Banner 的 HTML 文件 12.6.2.html：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/
xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>我的花店</title>
```

```
<link rel="stylesheet" href="css.css" type="text/css" />
</head>

<body>
<div class="banner">
<!-- left begin -->
 <div class="left">
     <div class="content_left">
         <p class="school_en">FOR MY LOVER</p>
         <p class="school_en"> </p>
         <p class="school_ch">真情永久</p>
         <p class="advertise">愿我们之间真情永久</p>
         <ul class="style_a">
             <li class="current"><a href="#">1</a></li>
             <li><a href="#">2</a></li>
             <li><a href="#">3</a></li>
             <li><a href="#">4</a></li>
         </ul>
     </div>
 </div>
<!-- left end -->
<!-- right begin -->
 <div class="right">
     <div class="content_right">
         <h4> </h4>
         <h4> </h4>
         <h4>红康乃馨 9 枝,香槟玫瑰 12 枝,配草内衬白色手揉纸,外朱红色手揉纸,朱红色蝴蝶结
</h4>
         <p> </p>
         <p class="bb">原价:188</p>
         <p> </p>
       <p class="aa">现价:168</p>
     </div>
 </div>
<!-- right end -->
</div>
</body>
</html>
```

(3) 代码在浏览器中运行的效果如图 12-2 所示。



图 12-2 花店 Banner 的运行效果

第13章 JavaScript

本章学习目标：

✍了解 JavaScript 的基础知识及语法规则。
✍能够熟练使用 JavaScript 编写程序。
✍掌握 JavaScript 中的事件和对象。

13.1 JavaScript 基础

JavaScript 是世界上最流行的编程语言，这门语言可用于 HTML 和 Web，更可广泛用于服务器、PC、笔记本电脑、平板电脑和智能手机等设备。

13.1.1 JavaScript 概述

JavaScript 是一种基于对象（Object-based）和事件驱动（Event-Driven）并具有安全性能的脚本语言，使用它的目的是与 HTML 超文本标记语言、Java 脚本语言（Java 小程序）一起实现在一个 Web 页面中链接多个对象，与 Web 客户交互，从而可以开发客户端的应用程序等。它是通过嵌入或调入在标准的 HTML 语言中实现的。

1. 特点

1）一种脚本编程语言

首先要解释一下什么是脚本语言。脚本语言是一种简单的程序，由一些 ASCII 字符构成，它直接利用记事本等文本编辑软件就可以开发完成，并且事先无须编译，只要利用适当的解释器就可以翻译并执行。

JavaScript 是一种脚本语言，它采用小程序段的方式实现编程，像其他脚本语言一样，JavaScript 同样是一种解释性语言，它提供了一个容易的开发过程。其基本结构形式与 C、C++、VB、Delphi 十分类似，但它不像这些语言需要先进行编译然后才能执行，而是在程序运行过程中被逐行地解释执行。它与 HTML 标识结合在一起，方便了用户的使用。

2）基于对象的语言

JavaScript 是一种基于对象的语言，这意味着它能运用自己创建的对象，因此，许多功能可以来自于脚本环境中对象方法的调用。

3）简单性

JavaScript 的简单性首先体现在它是一种基于 Java 基本语句和控制流之上的简单且紧凑的设计，从而对于学习 Java 是一种非常好的过渡；其次它的变量类型采用弱类型，并未使用严格的数据类型。

4）安全性

JavaScript 是一种安全的语言，它不允许访问本地的硬盘，且不能将数据存入到服务器上，不允许对网络文档进行修改和删除，只能通过浏览器实现信息浏览或动态交互，从而使得对数据的操作安全化。

5）动态性

JavaScript 是动态的，它可以直接对用户或客户的输入做出响应，无须经过 Web 服务程序。它对用户的响应是采用事件驱动的方式进行的。所谓事件驱动，是指在页面中执行了某种操作后产生相应的动作，例如按下鼠标、移动窗口、选择菜单等都可以视为事件，而当事件发生后可能会引起相应的事件响应。

6）跨平台性

JavaScript 与操作环境无关，只依赖于浏览器本身，只要计算机安装了支持 JavaScript 的浏览器，它就可以被正确执行，从而实现了“编写一次，走遍天下”的梦想。

综上所述，JavaScript 是一种描述性语言，它可以被嵌入到 HTML 的文件之中。它可以做到回应使用者的需求事件（如 form 的输入），而不用任何的网络回传资料，所以当一位使用者输入一项资料时，它不用经过传给服务端（Server）处理再传回来的过程，而直接可以被客户端（Client）的应用程序所处理。

2. JavaScript 可以用来做什么

JavaScript 虽然是一种简单的语言，但它的功能却很强大，具体如下：

1）制作网页特效

这几乎是所有 JavaScript 的初学者想学习 JavaScript 的第一个动机，例如光标动画、信息提示、动态广告面板、检测鼠标行为等。

2）提升使用性能

越是复杂的代码越需要耗费系统资源来执行，因为大部分的 JavaScript 程序代码都在客户端执行，在操作时完全不用服务器操心，这样网页服务器就可以将资源用在给客户提供更多、更好的服务上。例如申请会员要填的表单，JavaScript 的任务就是在数据送到服务器前进行有效性测试。

3）窗口动态操作

利用 JavaScript 可以自由地设计网页窗口的大小、打开与关闭等，甚至可以在不同窗口文件中互相传递参数。

3. 在 Web 中的应用

1）在 HTML 中嵌入 JavaScript

在 HTML 中通过标记<script>…</script>引入 JavaScript 代码，当浏览器读取到<script>标记时，解释执行其中的脚本。其中，可以将<script>标记放在头部<head>标

记中，此时 JavaScript 代码必须定义成函数形式，并在主体＜body＞标记内调用或通过事件触发。通常，放在＜head＞标记内的脚本在页面装载时同时载入，这样在主体＜body＞标记内调用时可以直接执行，提高了脚本的执行速度。当然，也可以将＜script＞标记放在主体＜body＞标记中，此时 JavaScript 代码可以定义成函数形式，在主体＜body＞标记内调用或通过事件触发，也可以在＜script＞标记内直接编写脚本语句，在页面装载时同时执行相关代码，这些代码执行的结果直接构成网页的内容。

2）链接 JavaScript 文件

在 Web 页中引入 JavaScript 程序的另一种形式是采用链接 JavaScript 文件的形式，如果脚本程序较长或者同一段脚本可以在若干个 Web 页中使用，则可以将脚本放在单独的一个 JavaScript 文件里，然后链接到需要它的 HTML 文件，这相当于将其中的脚本填入链接处。

如果要引用外部脚本文件，使用＜script＞标记的 src 属性来指定外部脚本文件的 URL。如果使用了＜script＞标记的 src 属性，则 Web 浏览器只使用在外部文件中的脚本，并忽略位于该＜script＞标记之间的任何脚本。

4. 编写工具

对于 JavaScript 脚本的编写，可以采用两种方式，一种是使用纯文本编辑器，另一种是使用专业化脚本编辑工具。

1）使用纯文本编辑器

使用纯文本编辑器（如 Windows 的记事本）来编写脚本是早期编程人员常用的方法。这种方法的优点是简单、易用；缺点是由于这种编辑器的主要用途是编辑纯文本，不具备对 JavaScript 语言的特性支持。因此，它多适用于脚本的少量编写和修改，当要进行大量的脚本编写和设计时，则需要专业化的脚本开发工具。

2）使用专业化脚本编辑工具

使用可视化工具（如 Frontpage、Dreamweaver 以及 Flash 等工具）可以十分容易地在 Web 页面中加入脚本，从而完成一些功能。这些工具是处理 JavaScript 的专业化开发工具，具有许多处理 JavaScript 特性的功能，如代码自动生成、语法敏感编辑、调试等，因此现在的开发人员经常使用这些工具进行 Web 程序的开发，以提高效率。必须注意的是，这些工具在自动生成有关的 JavaScript 代码时会加入一些冗余的代码，但这不会影响熟练的脚本编程人员对 Web 页中脚本的控制。

13.1.2 第一个 JS 程序

打开记事本，将以下代码录入记事本中，然后选择【文件】|【保存】命令，将文件类型选为【全部】，并将文件命名为 javascript1.html，浏览文件。注意，此处保存的文件扩展名必须加上。

```
<!-- javascript1.html -->
<html>
  <head>
  <title>第一个 JavaScript 实例</title>
```

```
    </head>
    <body>
        <script type = "text/javascript">
            document.write{"第一个 JavaScript 实例!};
        </script>
    </body>
</html>
```

13.1.3 编写与规则

1. 代码的编写与执行

JavaScript 是一种脚本语言。脚本语言是指在 Web 浏览器内由解释器解释执行的编程，在每次执行的时候解释器都会把程序代码翻译成可执行的格式。使用脚本语言编写的程序都是在脚本引擎装载 HTML 页面时解释执行的，脚本引擎是一个解释器，它是 Web 浏览器的一部分。一个包含翻译脚本的脚本引擎的 Web 浏览器称为脚本宿主，例如 Internet Explorer 就是 JavaScript 程序的脚本宿主。

JavaScript 既可以应用于客户端(浏览器端)，也可以应用于服务器端，只是在使用时有所区别，具体如下：

1）编程的区别

在客户端应用 JavaScript 时是把 JavaScript 代码通过<script>...</script>标记嵌入到 HTML 代码中；而在服务器端应用 JavaScript 时是把 JavaScript 代码通过<server>...</server>标记或通过<script language= "javascript" runat= "server">...</script>的方式嵌入服务器端的脚本或程序中。

2）执行方式的区别

客户端的 JavaScript 是浏览器通过解释的方式执行，而服务器端的 JavaScript 是通过编译的方式执行，且执行后将生成 *.web 类型的文件。

JavaScript 是一种解释语言，其源代码在发往客户端执行之前不需要经过编译，而是将文本格式的字符代码发送给客户端由浏览器解释执行，它是由上而下地执行，这种语言执行的弱点是容错性较差，如果一条执行不了，那么下面的语言也无法执行，而且由于一条一条地解释，速度较慢。

2. 语法规则

1）大小写

JavaScript 对大小写是敏感的，这意味着大写字母和相应的小写字母是不同的，如果在脚本中使用 result、Result 和 RESULT，每一个标识符都会被认为是不同的变量。大小写敏感适用于这种语言的各个方面，包括关键字、运算符、变量名、事件处理器、对象属性等。所有 JavaScript 的关键字都是小写的，例如使用 if 语句的时候，必须确保输入的是 if，而不是 IF，因为 JavaScript 使用了“驼山”命名的传统，许多方法和属性的使用混合大小写。比如，M 在 Document 对象的 lastModified 属性中必须是大写的，使用小写的 m 将会导致返回一个未被定义的值。

大小写敏感意味着用户在定义和使用变量时必须非常小心，在使用结构化语句（例如 if 和 while）的时候以及使用对象属性的时候同样如此，一个拼写上的错误将会导致整个脚本的含义都会发生变化，从而带来很多调试上的麻烦。

2）代码书写格式

下面是 JavaScript 脚本程序的几种基本格式：

（1）将 JavaScript 的程序代码放在<script>…</script>中。

```
<script>
document.write("Hello World!!!");
</script>
```

（2）在<script>标记中加上 language 属性，指定使用哪一种语言。

```
<script language = "JavaScript">
document.write("Hello World!!!");
</script>
```

（3）在标记 language 属性外进一步指明 JavaScript 的版本号。

```
<script language = "JavaScript1.0">
document.write("Hello World!!!");
</script>
```

（4）如果基于模块化考虑，可以将可能重复使用的程序代码独立成一个外部文件，只要以.js 为扩展名，例如 hello.js，利用 src 属性就可以将该文件的内容包含进来。

```
<script src = "hello.js">
</script>
```

说明：“document.write("Hello World!!!");”必须保存为一个外部文件 hello.js。

（5）为避免客户端使用了版本较低的浏览器，造成对 JavaScript 不完全支持，从而可能出现许多错误，可以将所有的程序代码用 HTML 的注释标记括起来，让这些低版本的浏览器视而不见。

```
<script language = "JavaScript">
<!-- 以下程序隐藏起来,让旧的浏览器视而不见
document.write("Hello World!!!");
//程序代码隐藏到此为止 -->
</script>
```

（6）使用 JavaScript 协议。

```
<a href = "JavaScript:alert('Hello World!!!')">请单击</a>
<a href = "#" onclick = "alert('Hello World!!!')">请单击</a>
<a href = "JavaScript://" onclick = "alert('Hello World!!!')">请单击</a>
```

3）保留字

基本上，每一种编程语言都有它的保留字，JavaScript 也有它自己的保留字，JavaScript 还有一些保留字不能在标识符中使用。保留字对 JavaScript 语言有特殊的含义，它们是语

言语法的一部分，使用保留字在加载脚本的时候将产生编译错误。保留字如表 13-1 所示。

表 13-1 JavaScript 的保留字

保留字				
break	case	catch	continue	debugger
default	delete	do	else	false
finally	for	function	if	in
instanceof	new	null	return	switch
this	throw	true	try	typeof
var	void	while	with	

JavaScript 还有一些留给将来使用的保留字，这些保留字不是现在 JavaScript 语言的一部分，它们是为将来的使用保留的。JavaScript 未使用的保留字如表 13-2 所示。

表 13-2 JavaScript 未使用的保留字

未使用的保留字				
abstract	double	goto	native	static
boolean	enum	implements	package	super
byte	export	import	private	synchronizd
char	extends	int	protected	throw
class	final	interface	pubilc	transients
const	float	long	short	volatile

4）分号和语句结束符

JavaScript 区分字母的大小写，语句以分号“；”作为结束符，分号意味着 JavaScript 语句的结束，也可以将多个语句写在同一行中，用分号将它们隔离开来。

13.2 JavaScript 程序

JavaScript 程序由语句、语句块、函数、对象、方法、属性等构成，程序结构可分为顺序、分支和循环 3 种基本结构。

13.2.1 语句和语句块

JavaScript 语句是发送给浏览器的命令，这些命令的作用是告诉浏览器要做的事情。例如下面语句的作用是告诉浏览器在页面上输出“我是 JavaScript 程序!”。

```
document.write{"我是 JavaScript 程序!};
```

多行 JavaScript 语句可以组合起来形成语句块，语句块以左花括号“{”开始，以右花括号“}”结束。下面的语句块向网页输出一个标题以及两个段落。

```
< script type = "text/JavaScrip">
{
document.write {"< h1 >标题 1 </hl >");
```

```
document.write{"<p>这是段落 1</p>");
document.write{"<p>这是段落 2</p>");
}
</script>
```

13.2.2 代码

JavaScript 代码是由若干条语句或语句块构成的执行体，在以下代码中第 2～7 行由语句和语句块构成的就是 JavaScript 代码。

```
1 <script type = "text/javascript">
2   var color = "red";
3   if(color == "red")
4 {
5   document.write("颜色是红色!");
6   alert("颜色是红色!");
7 }
8 </script>
```

13.2.3 消息对话框

JavaScript 中的消息对话框分为告警框、确认框和提示框 3 种。

1. 告警框

alert()函数用于显示带有一个图标、一条指定消息和一个【确定】按钮的告警框。

基本语法：

```
alert(message);
```

说明：message 参数用于显示弹出对话框上的纯文本(非 HTML 文本)。

2. 确认框

confirm()方法用于显示带有一个图标、指定消息和【确定】及【取消】按钮的对话框。

基本语法：

```
confirm(message);
```

说明：message 参数用于显示弹出对话框上的纯文本(非 HTML 文本)。

如果用户单击【确定】按钮，则 confirm()返回 true；如果单击【取消】按钮，则 confirm()返回 false；在用户单击【确定】按钮或【取消】按钮把对话框关闭之前，它将阻止用户对浏览器的所有操作。在调用 confirm()时，将暂停对 JavaScript 代码的执行，在用户做出响应之前不会执行下一条语句。

3. 提示框

prompt()方法用于提示用户在进入页面前输入某个值。

基本语法：

```
prompt(text,defaultValue)
```

说明：text 用于设置提示信息；defaultValue 用于设置默认的输入值。

如果用户单击提示框中的【取消】按钮，则返回 null；如果用户单击【确定】按钮，则返回在文本输入框中输入的值；在用户单击【确定】按钮或【取消】按钮把对话框关闭之前，它将阻止用户对浏览器的所有操作。在调用 prompt()时，将暂停对 JavaScript 代码的执行，在用户作出响应之前不会执行下一条语句。

13.2.4 注释

JavaScript 提供了两种类型的注释，即单行注释和多行注释。单行注释使用“//”作为注释标记，可以单独一行，或跟在代码末尾放在同一行中，“//”后为注释内容部分。当注释行数较少时适合使用单行注释，如果注释行数较多，则需要在每行的开头加“//”，比较麻烦，此时应使用多行注释。多行注释可以包含任意行数的注释文本，以“/ * ”标记开始、以“ * /”标记结束，在两个标记之间所有的内容都是注释文本，所有注释的内容都将被浏览器忽略，不影响页面效果和程序的执行，对以后用户阅读和维护程序十分方便。

使用注释可以防止代码执行，注释的作用是为代码添加阅读说明，但有时也会用来屏蔽某些语句行的执行，对程序调试非常有用。例如：

```
<!-- javascript.html -->
<html>
  <head>
  <title>注释使用实例</title>
  </head>
  <body>
  <script type = "text/javascript">
  //这是单行注释
  /*
  这是多行注释
  可以包含多行内容
  */
  //alert("此语句不执行!");
  alert("此语句执行了!");          //执行时弹出告警消息框
  </script>
  </body>
</html>
```

13.3 变量、数据类型和表达式

在 JavaScript 中变量是存储信息的容器，它拥有动态类型，这意味着相同的变量可用作不同的类型，而表达式是解释器能够对变量进行计算的语句。

13.3.1 变量

1. 变量的声明和初始化

JavaScript 变量是一个存储或者表示数据的名称，可用来存储和表示各种数据类型的数据，并且这些值在程序运行期间是可以改变的。JavaScript 是一种弱数据类型，统一使用关键字 var 声明，JavaScript 会在需要时自动对不同的数据类型进行转换。

语法：

```
var 变量名[ = 初值][,变量名[ = 初值]…]
```

说明：var(variant)是关键字，在声明时至少要有一个变量，每个变量要起一个合适的名称，变量的命名应该符合规范；可以同时声明多个变量，多个变量之间用逗号“,”分隔，并可以在声明变量的同时进行赋值。

每条声明语句均需要以“;”结束，这是一个好习惯。

例如：

```
var xl,yl;
var str;
```

2. 变量的命名

创建合法的变量必须满足下面几个规则：

- 第一个字是 ASCII 字母(大小写均可)或者下划线，但不可以是数字。
- 变量必须由字母、数字和下划线组成。
- 变量不可以是保留字。

3. 变量的作用域

所谓变量的作用域就是变量发生作用的范围。变量在声明之后，只在自己的作用域中有效，在变量的作用域外使用变量将导致解释器报错。

```
var x;
function FirstFuntion()
{
  var y = 2;
}
x = y;     //错误,在变量 y 的作用域外使用变量
```

在上面这个例子中有一个简单的函数 FirstFuntion()，这个函数声明并初始化了一个变量 y，在函数结束后，将 y 的值又赋给了 x，这时 y 已经在它的作用域外，使用变量 y 将导致解释器报错。

13.3.2 数据类型

数据类型是每一种计算机语言的重要基础，JavaScript 中的数据类型可以分为字符型、数值型、布尔型、Null、Undefined 和对象 6 种。

1. 字符型(String)

字符型数据又称为字符串,由若干个字符组成,并且需要用单引号(')或双引号(")封装起来(在 JavaScript 中,使用单引号和双引号的效果是一样的)。下面的例子列举了正确和错误使用字符型数据的两种情形:

```
"Tiger",'JavaScript 字符串'     (正确)
'document","你好'       (错误,单引号、双引号不配)
```

在使用字符串的过程中,有时会遇到一种情况:在 4 个字符串中需要使用单引号或双引号。正确的方法是在用双引号标记的字符串中加入引用字符时使用单引号,在用单引号标记的字符串中加入引用字符时使用双引号,即保证一个字符串的开头和结尾使用同一种引号,而字符串内使用另一种引号。下面给出了正确的用法:

```
"热烈欢迎参加'JavaScript 技术'研讨的专家"
```

2. 数值型(Number)

与其他编程语言类似,JavaScript 中最基本的一种数据类型是数值型,该类型可分为整型、浮点型、内部常量以及特殊值。

(1) 整型:例如 100、—300、0 等都是整数。整数除了用十进制表示外,还可以用八进制和十六进制的方式表示。使用 0 开头的整数是八进制整数,如 017、—010 等都是合法的八进制整数;使用 0x 开头的整数是十六进制整数,如 0x10、0x1a3d 都是合法的十六进制整数。

(2) 浮点型:3.4、—540.66 都是浮点型数值,浮点数还可以采用科学记数法表示,如 2.3E12。

(3) 内部常量:JavaScript 中常用的内部常量如表 13-3 所示。

表 13-3 JavaScript 中的内部常量

常 量	说 明	常 量	说 明
Math. E	自然数	Math. LN2	2 的自然对数
Math. PI	圆周率	Math. LN10	10 的自然对数
Math. SQRT2	2 的平方根	Math. LOG2E	以 2 为底的 e 的对数
Math. SQRT1-2	1/2 的平方根	Math. LOG10E	以 10 为底的 e 的对数

(4) 特殊值:JavaScript 中的特殊值如表 13-4 所示。

表 13-4 JavaScript 中的特殊值

特 殊 量	说 明
Infinity	无穷大
Number. NaN	非数字值(Not a Number)
Number. MAX_VALUE	可表示的最大的数
Number. MIN_VALUE	可表示的最小的数
Number. NEGITIVE_INFINITY	负无穷大,溢出时返回该值
Number. POSITIVE_INFINITY	正无穷大,溢出时返回该值

3．布尔型（Boolean）

布尔型是一种只含有 true 和 false 两个值的数据类型，通常来说，布尔型数据表示“真”或“假”。在实际应用中，布尔型数据常用在比较、逻辑等运算中，运算的结果往往是 true 或者 false。例如 3>2 的比较结果是 true，而 6==4 的比较结果是 false。此外，布尔型变量还常用在控制结构的语句中，例如 if 语句等。

在 JavaScript 中，通常采用 true 和 false 表示布尔型数据，但也可将它们转换为其他类型的数据，例如可将值为 true 的布尔型数据转换为整数 1，而将值为 false 的布尔型数据转换为整数 0，但不能用 true 表示 1 或用 false 表示 0。

4．Null

在 JavaScript 中，Null 是一种特殊的数据类型，也称为空类型，此类型只有一个值，即 Null，表示“无值”，什么也不表示。Null 除了表示 Null 类型的数据外，也可用在表示其他类型的数据中，如对象、数组等，当变量不再使用时，将它赋值为 Null，以释放存储空间。

5．Undefined

在 JavaScript 中，Undefined 也是一种特殊的值，是指变量创建之后还没有赋值之前所具有的值，其返回值就是 Undefined。它与 Null 值的不同之处在于 Null 值表示已经给变量赋值，只不过赋的值是“无值”，而 Undefined 表示变量不存在或者没有赋值，如果使用未定义的变量也会显示 Undefined，但通常使用未定义的变量会造成程序错误。

6．对象（Object）

在 JavaScript 中，除了数值型、字符型和布尔型等这些基本的数据类型外，还有一种复合的数据类型，称为对象。对象是属性和方法的集合，对象的属性可以是任何类型的数据，包括数值、字符、布尔型，甚至可以是另一种类型的对象，而方法是一个定义在对象中的函数，用于实现特定的功能。

在 JavaScript 中定义了多个对象，例如 Data、Window、Document 等。

13.3.3 运算符与表达式

1．表达式

表达式是解释器能够对变量进行计算的语句，表达式主要由任何合法类型的变量和运算符组成，下面是几个表达式的例子：

```
var x = 4;
x = x + 4;
var y = x;
 y = y + 6;
```

分号代表了一个表达式的结束。使用分号可以在同一行中包括多个表达式，例如：

```
var x = 4; var y = x; var y = y + 1;
```

在有些时候，分号是可以省略的，例如本节的第一个例子也可以写成：

```
var X = 4
x = x + 4
var y = x
 y = y + 6
```

这是因为这些表达式是被分行符隔开的。但是如果下行表达式可以看作是上行表达式的扩展，则必须插入分号，一个很典型的例子是 return 关键字的使用：

```
return
x
```

将被解释器看作

```
return;
x;
```

而不是期望中的

```
return x
```

这是因为 return 关键字的参数是可选的，它的含义是返回运行结果，在遇到换行符的情况下，解释器将不返回任何参数，只是简单地执行返回操作。

如果需要使用一组表达式表示一个特殊的条件或者结果，可以用一个大括号"{}"将这组表达式放置在一起，被大括号"{}"放置在一起的一组表达式被称为一个块，例如：

```
if(x = 4)
{
  x = x + 4;
  var y = x;
}
```

在这段代码中，大括号中的代码只有在"x＝4"成立的时候才会被执行，并且这两个表达式要么都被执行，要么都不执行。这里需要注意的是变量 y 的作用域，在大括号外使用变量 y 将出现错误。

注意，正确地使用块将有助于提高程序的可读性。

2. 算术运算符和赋值运算符

运算符用来连接表达式中的变量，并对这些变量进行各种操作。

JavaScript 支持所有数学中的算术运算符，例如加(＋)、减(－)、乘(*)、除(/)等。表 13-5 列出了算术运算符的基本功能。

表 13-5 JavaScript 的算术运算符

运 算 符	名 称	基 本 功 能
＋	加法运算符	求＋号两边变量的和值，同数学符号＋
－	减法运算符	求－号两边变量的差值，同数学符号－
*	乘法运算符	求 * 号两边变量的乘积，同数学符号 *

续表

运　算　符	名　　称	基 本 功 能
/	除法运算符	将/号前变量均分为/后变量的分数，同数学符号/
%^	模运算符	求^左边的变量除以右边变量后的余数
++	递加运算符	将++号左边或右边的变量加1
--	递减运算符	将--号左边或右边的变量减1

需要特别说明的是，当“+”两边的变量是字符串时，“+”不做加法运算，而是做连接操作，即将“+”两边的字符串连接起来，例如：

```
var x = "hello " + "world.";
```

变量 x 的输出结果是：

```
hello world.
```

当“+”作为字符串连接时也允许变量和字符串结合在一起，例如：

```
var x = "hello world.";
var y = x + " begin to study javascript.";
```

变量 y 的输出结果为：

```
hello world. begin to study javascript.
```

当“+”同时遇到字符串型和数值类型时，也许会碰到没有预料到的结果，这是因为如果“+”在执行时遇到了一个字符串，它会将今后遇到的任何类型数值或变量当成字符串进行处理。

赋值运算符“=”的作用是将某个值赋给一个变量，这种赋值的操作既可以是一对一的，也可以是同时进行的，例如：

```
var x = "7 is a number";
var a = b = c = 7;
```

3. 比较运算符和逻辑运算符

比较运算符的作用是判断表达式的真假，因此比较运算符返回的结果是布尔型的，JavaScript 中常见的比较运算符见表 13-6。

表 13-6　JavaScript 比较运算符

符　　号	含　　义	举　　例	结　　果
==	等于	6-4==2	true
! =	不等于	2+3! =1	true
>=	大于等于	2+3>=5	true
<=	小于等于	3+1<=4	true
>	大于	2+1>3	false
<	小于	2+1<3	false
===	等于(同一类型)	2+1==='2+1'	false
! ==	不等于(同一类型)	1+2! ==3	true

特别需要注意的是“＝”和“＝＝”的区别，“＝”是赋值运算符，代表将某个值赋给某个变量；“＝＝”是逻辑运算符，被用来判断两个量是否相等。例如：

```
y = (x == 2015);
```

“x＝＝2015”是一个逻辑判断表达式，表达式的运算结果是 true 或者是 false，在解释器计算出这个表达式的值后将值赋给 y。

此外，含有“＝＝＝”的比较表达式为真的条件是“＝＝＝”两边的变量不仅数值上必须相等，数据类型也必须是一样的。

逻辑运算符的作用是将比较表达式组合起来作为判断的依据，因此逻辑运算符最常见于 for 语句的条件中，用来控制流程。表 13-7 列出了 JavaScript 的逻辑表达式。

表 13-7　JavaScript 的逻辑表达式

符　号	名　称	含　义
!	逻辑非	如果操作数是 true，则返回 false；反之亦然
&&	逻辑与	如果两个操作数都是 true，则退回 true；否则返回 false
||	逻辑或	两个操作数中任何一个是 true，则返回 true；否则返回 false

4. 位运算符和条件运算符

位运算符是对字符位数进行控制的运算符。对两个十进制的整数进行位运算时，JavaScript 先将这两个数转换成 32 位整数，然后在位运算前将这两个数字转换成位字符。JavaScript 的位运算符如表 13-8 所示。

表 13-8　JavaScript 的位运算符

符　号	含　义	举　例	结　果
<<	按位左移	12<<3	96
>>	按位右移	12>>3	1
>>>	无符号右移	12>>>3	1
—	按位非	—12	—13
&	按位与	12&3	0
^	按位异或	12^3	15
|	按位或	12|3	15

这里以“12＜＜3”为例，将 12 转换为二进制数后为 00001100，按位左移 3 位为 01100000，转换成十进制数是 96，因此 1<<3＝96。

再看“12&3”，12 转换为二进制数是 00001100，3 转换为二进制数为 00000011，看下列运算

```
  00001100
& 00000011
  00000000
```

当上、下两个数字都是 1 时，结果是 1，否则是 0，所以 12&3＝0。

按位与、按位异或和按位或的运算规则如表 13-9 所示。

表 13-9　JavaScript 的逻辑按位与、按位异或和按位或运算规则

第一个字节	第二个字节	按位与(&)	按位异或(^)	按位或(|)
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

条件运算符是 JavaScript 中唯一的一个三元运算符。所谓三元运算符就是在使用运算符时必须有 3 个变量存在,条件运算符用符号"?"表示,其使用方法如下:

```
a?b?c
```

a、b 和 c 是 3 个表达式,当表达式 a 成立时,取表达式 b 的值,否则取表达式 c 的值。例如:

```
var x = (a > 10)?10:20;
```

这个运行代码声明了一个变量 x,x 的值是由 a 的值决定的,如果 a 的值大于 10,那么为变量 x 赋值 10,否则给变量 x 赋值 20。

5. 其他运算符

在 JavaScript 中除了上述运算符外,还有一些其他运算符,如表 13-10 所示。

表 13-10　其他运算符

运　算　符	操 作 说 明
,	逗号运算符
new	新建对象运算符
delete	删除对象运算符
typeof	类型运算符

1) 逗号运算符(,)

逗号运算符是先计算第一个表达式的值,再计算第二个表达式的值,运行结果为第二个表达式的值。

例如:

```
var rs;
rs = (3 + 5,10 * 6);
```

这个例子先计算第一个表达式 3+5 的值为 8,再计算第二个表达式 10 * 6 的值为 60,最后将第二个表达式的值 60 赋给变量 rs。

2) 新建对象运算符(new)

新建对象运算符用于创建 JavaScript 对象实例或数组。例如:

```
var obj  =  new Object();
```

```
var date = new Date();
var array = new Array();
```

在上例中,第一个语句是创建一个 Object 对象,对象名为 obj;第二个是创建一个 Date 对象,对象名为 date;第三个是创建一个 Array 对象,对象名为 array。

3) 删除运算符(delete)

删除运算符用于删除一个对象的属性或某个数组的元素。例如:

```
delete array(30);
delete obj.height;
```

在上例中,第一个语句是删除 array 数组中下标为 30 的元素(第 31 元素),第二个是删除对象 obj 的 height 属性。

4) 类型运算符(typeof)

类型运算符是一个一元运算符,其操作数可以是任意类型,运算结果返回一个表示操作数类型的字符串。例如:

```
typeof(300);
typeof("Hello");
```

在上例中,第一个运算的结果为 Number,第二个运算的结果为 String。

类型运算符的具体规则如表 13-11 所示。

表 13-11 类型运算符的运算规则

数据类型	运算结果	数据类型	运算结果
数字型	Number	数组	
字符型	String	函数	
布尔型	Boolean	Null	
对象	Object	未定义	undefined

13.4 控制结构

在 HTML 基础上,使用 JavaScript 可以开发交互式 Web 页面,例如可以在线填写各类表格、联机编写文档并发布等。JavaScript 的出现使得网页和用户之间实现了一种实时性的、动态的、交互性的关系,使网页包含更多活跃元素和更加精彩的内容,这也是 JavaScript 与 HTML DOM 共同构成 Web 网页的行为。在网页设计中,JavaScript 的主要作用是实现内容与行为的分离,而要设计交互式的页面必须编写相应的脚本程序。程序是专门用来解决某一问题的特定代码。

JavaScript 程序设计分为两种方式,即面向过程和面向对象的程序设计,每种方式都是对数据结构和算法的描述。在所有编程语言中,程序的结构有 3 种,分别为顺序结构、分支结构和循环结构,任何复杂的算法均可以使用这 3 种结构来表达。

13.4.1 顺序结构

顺序结构是程序设计中最常用、最基本的一种程序结构，是按照语句出现的顺序从第一条语句开始一步一步逐条执行，直到最后一条语句。

13.4.2 分支结构

在编写代码时经常需要根据不同的条件完成不同的行为，用户可以在代码中使用条件语句来完成这一任务，在 JavaScript 中可以使用下面 4 种形式的分支结构语句。

- 单 if( ){语句块}语句：在条件成立时执行语句块。
- 双 if( ){语句块 1}else{语句块 2}语句：在指定条件成立时执行语句块 1，不成立时执行语句块 2。
- 多 if( ){语句块 1}else if( ){语句块 2}…else{语句块 *n*)语句：在指定条件成立时执行语句块 1，否则判断第 2 个条件，如果成立执行语句块 2，依此类推，直到所有条件均不成功时执行语句块 *n*。
- 多分支 switch( ){}语句：根据变量或表达式的值与 case 常量匹配的情况选择其中一个分支执行。

1. if 语句

if 语句是分支条件语句，即根据一个条件来控制程序执行的流程。

语法：

```
if(表达式){
  条件为真时执行代码;
}
```

说明：

if 语句的小括号中表达式的结果类型必须是布尔型，即 true 或 false，当值为 true 时执行大括号中的代码，否则跳过大括号中的代码执行大括号后面的代码。if 语句的流程图如图 13-1 所示。

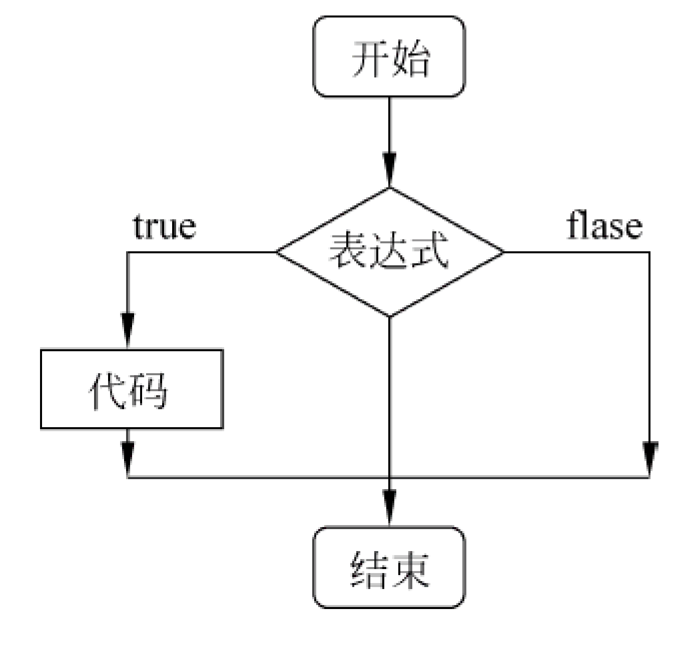


图 13-1　单分支 if 语句流程图

2. if…else 语句

if…else 语句是双分支条件语句，即根据一个条件来控制程序执行的流程。

语法：

```
if(表达式){
  条件成立时执行代码 1
}else{
  条件不成立时执行代码 2
}
```

说明：

if...else 语句的小括号中表达式的结果类型必须是布尔型，即 true 或 false，当值为 true 时执行代码 1，否则执行代码 2。if...else 语句的流程图如图 13-2 所示。

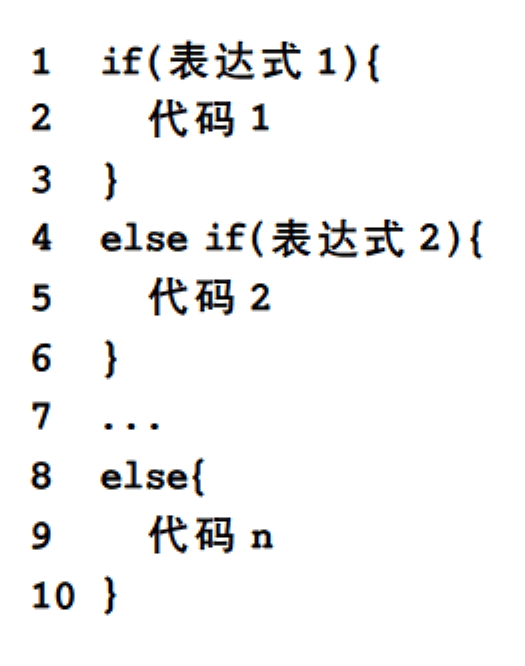


图 13-2 双分支 if 语句流程图

3. 多重 if...else 语句

if...else if...else 语句是多条件、多分支语句，可根据两个以上的条件来控制程序执行的流程。

语法：

```
if(表达式 1){
   代码 1
}
else if(表达式 2){
   代码 2
}
...
else{
   代码 n
}
```

说明：

在多重 if...else 语句中，if 以及多个 else if 后面的小括号内的表达式的值为布尔类型，在程序执行时，按照该语句中表达式的顺序首先计算表达式 1 的值，如果计算结果为 true，则执行代码 1，执行完结束 if...else if...else 语句；如果计算结果为 false，则继续计算表达式 2 的值，依此类推，假设第 m 个表达式的值为 true，则执行紧跟的代码 m，并结束 if...else if...else 语句的执行，否则继续计算 $m+1$ 个表达式的值，如果所有表达式的值都为 false，则执行关键字 else 后面的代码 n，结束 if...else if...else 语句的执行，其执行流程如图 13-3 所示。

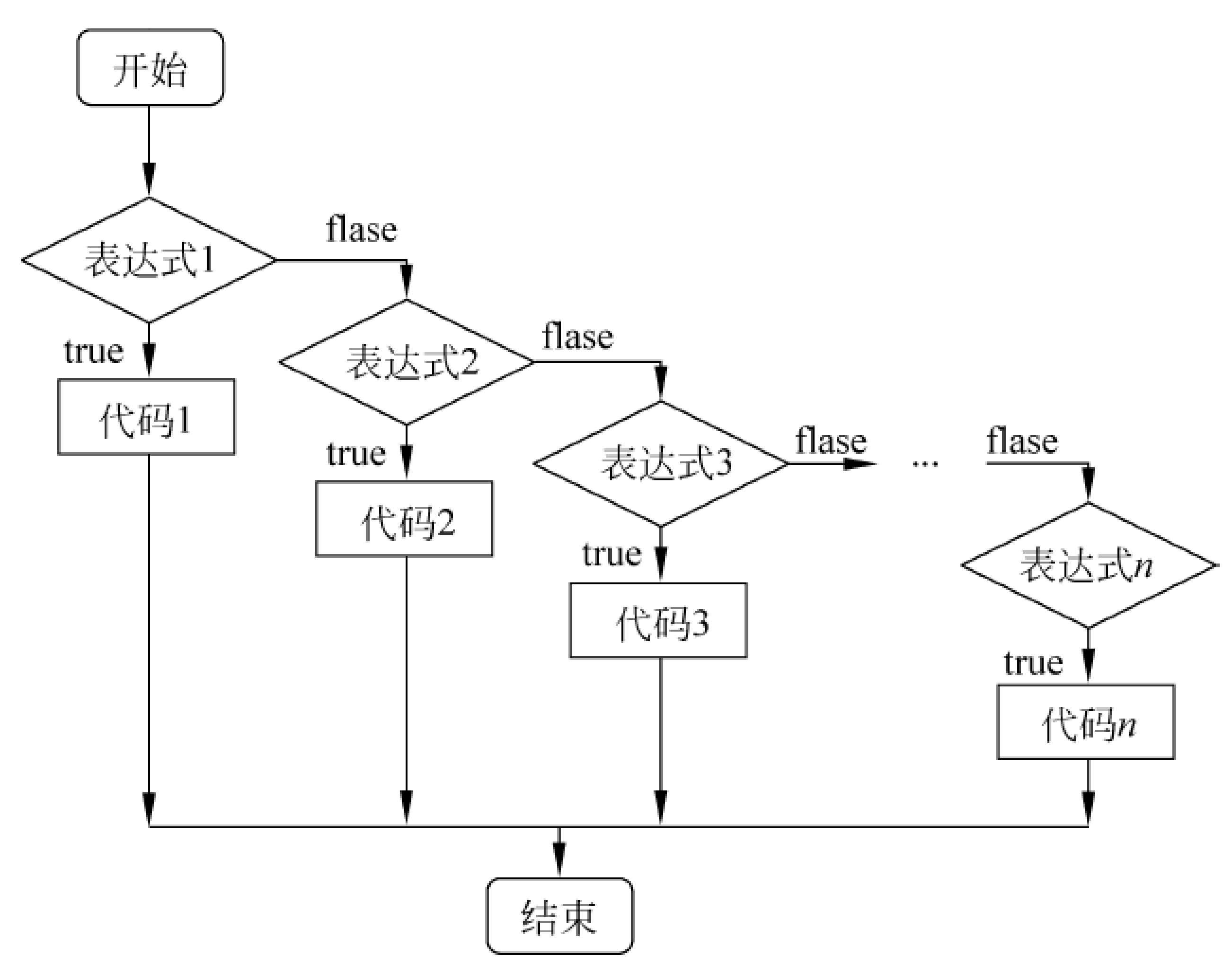


图 13-3 多重 if...else 条件语句流程图

4. switch 语句

switch 语句是单条件、多分支语句，它可以通过判断一个条件完成对程序多个分支的控制，比 if...else 的使用更方便、结构更清晰。

语法：

```
switch(表达式){
case 常量 1:
  {代码 1
  }break;
case 常量 2:
  {代码 2
  }break;
...
case 常量 n:
  {代码 n
  }break;
default
  {代码 n+1}
}
```

说明：

在执行 switch 语句时，首先计算变量或表达式的值，然后查找和这个值匹配的 case 常量，如果找到了匹配的常量，执行后面的语句块，否则执行 default 后面的语句块。

在上面的语法格式中，每个 case 语句块的后面都有一个 break 语句，其作用是终止 switch 语句的执行，继续执行 switch 下面的语句。如果没有这个 break 语句，那么 switch 语句会从和表达式的值匹配的 case 常量开始，依次执行后面所有的代码，直到 switch 语句的结尾部分。

13.4.3 循环结构

在实际问题中还有一种情况，那就是需要重复执行一组语句，直到达成目标。例如将一个班级中所有同学的名字输出到网页上，不能在页面中重复写 n 行相同的代码去输出所有同学的名字，这种情况使用循环结构可以解决，JavaScript 提供了 for、while、do...while、for...in 等多种循环。

1. for 循环

for 循环是一种结构简单、使用频率高的循环控制语句，作用是有条件地重复执行一段代码。

语法：

```
for(表达式 1;表达式 2;表达式 3)
{
  需要循环执行的代码;
}
```

说明：

for 是 for 语句的关键字，for 关键字后面的一对小括号“( )”不可省略，括号中用分号“;”分隔 3 个表达式，表达式 1 是初始表达式，在循环开始前执行，一般用来定义循环变量；表达式 2 是判断表达式，就是循环条件，必须为布尔型数据的表达式，当表达式的结果为 true 时循环继续执行，否则循环结束；表达式 3 是循环表达式，在每次循环执行后都被执行，作用是修改循环变量，然后进行判断表达式的计算，决定是否继续下一次循环。

花括号“{}”内的代码为循环体，当循环体只有一条语句时花括号可以省略。

for 语句的执行规则如下：

(1) 计算初始表达式的值，完成循环的初始化工作。

(2) 计算判断表达式的值，若判断表达式为 true，跳转到步骤(3)，否则跳转到步骤(4)。

(3) 执行循环体代码，然后计算循环表达式的值，以改变循环变量，跳转到步骤(2)。

(4) 结束 for 语句的执行。

for 语句的执行流程如图 13-4 所示。

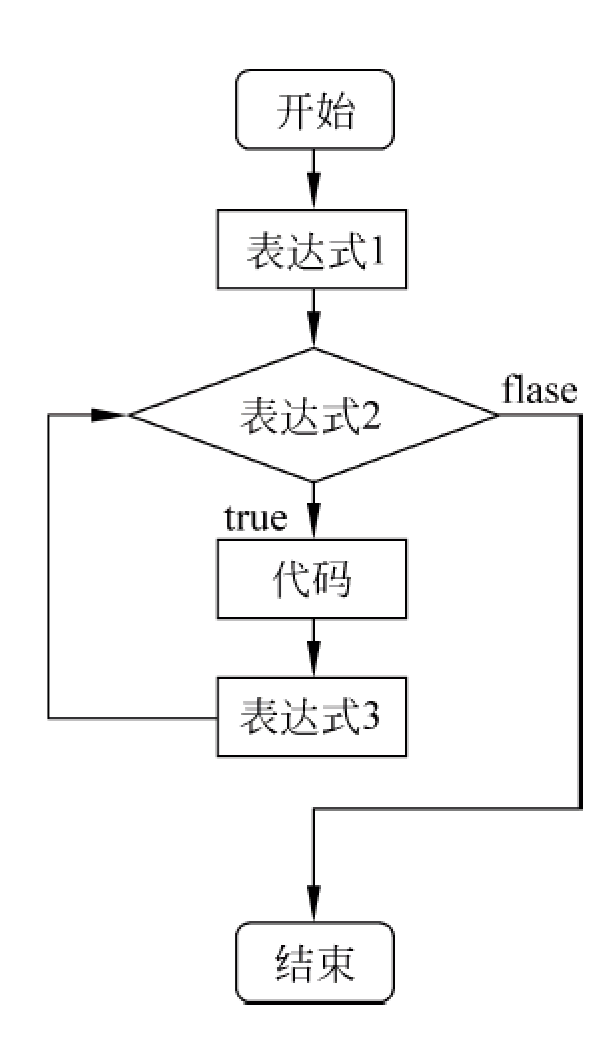


图 13-4 for 语句流程图

2. while 循环

while 循环是 JavaScript 中最基本的循环控制语句之一，其作用是有条件地重复执行某一段代码。

语法：

```
while(表达式)
{
   需要循环执行的代码;
}
```

说明：

while 语句由关键字 while、一对小括号“( )”中的表达式和一对花括号“{ }”中的代码组成，代码称为循环体，表达式称为循环条件。

while 语句的执行规则如下：

(1) 计算表达式的值，如果值为 true，跳转到步骤(2)，否则跳转到步骤(3)。

(2) 执行循环体代码，跳转到步骤(1)。

(3) 终止 while 语句的执行。

在使用 while 循环时需要注意以下几个问题：

- 在 while 循环之前必须完成循环变量的初始化工作。
- 使用“{}”将循环体语句括起来，即使是一条语句也要使用花括号。
- 在循环体语句中必须含有循环正常退出的语句，即循环控制语句，以避免发生死循环。

由于 while 循环中只有一个表达式，不像 for 循环有 3 个表达式，所以完成同样的工作需要在其他地方进行处理，如变量的初始化必须移到 while 循环开始之前；其次是循环表达式的工作必须移到 while 循环体内执行，只有这样 while 循环才能完成与 for 循环同样的功能。

while 语句的执行流程如图 13-5 所示。

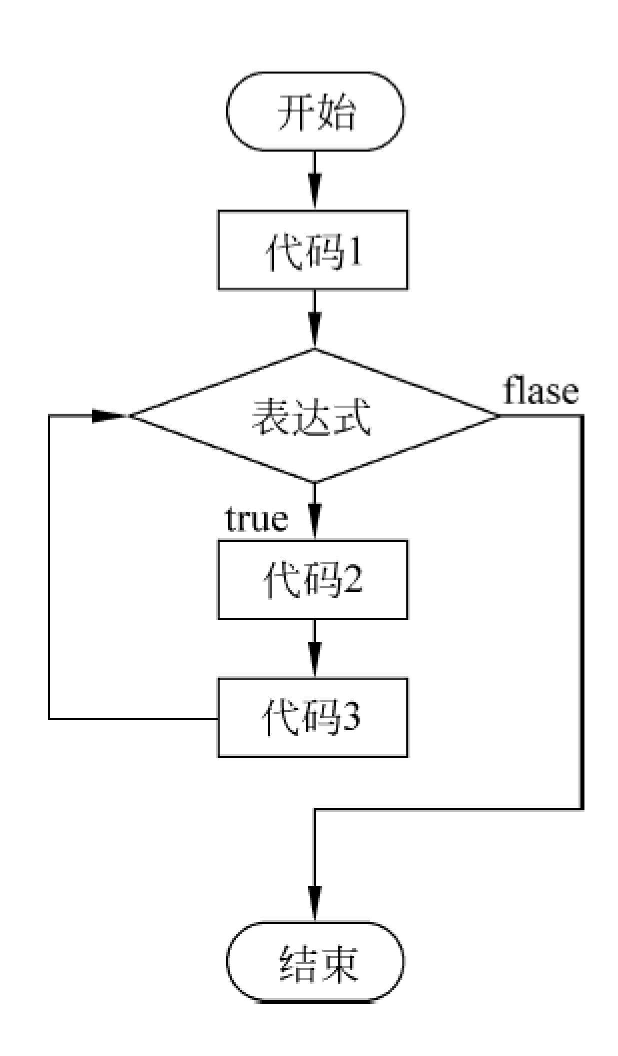


图 13-5 while 语句流程图

3. do…while 循环

do…while 循环和 while 循环非常类似，也用于重复执行某一段代码，它们的不同在于 while 循环的条件表达式位于 while 循环的头部，而 do…while 循环的条件表达式位于 do…while 循环的尾部，因此 while 循环总是先检测条件表达式是否成立，如果成立才执行循环体代码，而 do…while 循环先执行一次循环体内的代码，再判断条件表达式是否成立，如果成立继续执行循环体语句，否则结束循环。

语法：

```
do{
  需要循环执行的代码;
}while(表达式)
```

说明：

与 while 循环一样，在使用 do…while 循环时需要注意以下几个问题：

- 在使用 do…while 循环之前必须完成循环变量的初始化工作。
- 使用“{}”将循环体语句括起来，即使是一条语句也要使用花括号。
- 在循环体语句中必须含有使循环正常退出的语句，即循环控制语句，如 i++，避免发生死循环。

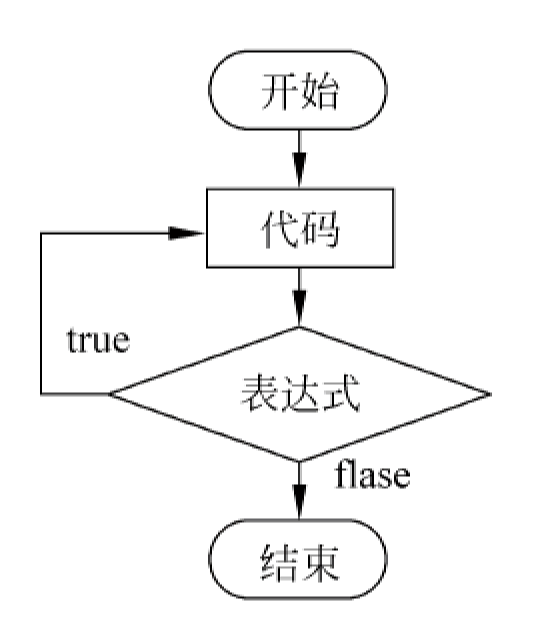


图 13-6 do…while 语句流程图

do…while 循环的执行规则如下：

(1) 执行循环体代码。

(2) 计算表达式的值，如果值为 true，跳转到步骤(1)，否则跳转到步骤(3)。

(3) 终止 do…while 语句的执行。

do…while 语句的执行流程如图 13-6 所示。

4. for…in 循环

在 JavaScript 中，除了 for 语句可以用于控制循环结构以外，还有另一种形式的 for 语句，主要用于数组、集合对象的遍历，也需要循环执行某一段代码，即 for…in 循环。

语法：

```
for(variable in object)
{
  需要循环执行的代码;
}
```

说明：

variable 可以是一个变量名、数组元素或对象属性，object 可以是一个对象名，或者计算结果为对象的表达式。for…in 循环将对 object 对象的每一个属性或每一个元素都执行一

次循环；在循环过程中，首先将 object 对象的一个属性名作为字符串赋给变量 variable，这样在循环体内就可以通过变量 variable 访问对象属性了。

5. 循环的嵌套

在一个循环内又包含另一个完整的循环结构，称为循环嵌套，嵌套其他循环就构成了多重循环结构。

6. break 和 continue 语句的作用

在正常的循环结构中，每次循环都是从满足条件开始直到不满足条件结束，也就是说必须完整地执行完所有的循环。但在实际问题中有时并不需要完整地执行完所有循环才结束程序，可能会遇到一些需要提前中止或跳过某些循环的情况，这时需要使用 break 和 continue 语句来解决实际问题。

在循环体中，break 语句的作用是立即结束循环，跳转到循环后面的语句，而不管原来的循环还有多少次，都不会再执行。

在循环体中，continue 语句的作用是结束本次循环，本次循环后面的所有语句都不会执行，直接进入下一次循环，直到循环结束。

13.5 函数

函数就是结构化的可重用的代码段，是用来完成一个特定功能的程序代码。函数只需要定义一次，可以多次使用，从而提高程序代码的复用率，既减轻了开发人员的负担，也降低了代码的重复度。

JavaScript 函数分为系统内部函数和系统对象定义的函数及用户自定义函数。函数需要先定义后使用，JavaScript 函数一般定义在 HTML 文件的头部 head 标记或外部 JS 文件中，而函数的调用可以在 HTML 文件的主体 body 标记中的任何位置。

13.5.1 常用系统函数

在 JavaScript 中有许多预先定义的系统内部函数和对象定义的函数，如 document.write()就是其中之一。这些预定义的系统函数大多存在于预定义的对象中，例如 String、Date、Math、Window 及 Document 对象中都有很多预定义的函数，只有熟练地使用这些函数才能充分发挥 JavaScript 强大的功能，简洁、高效地完成程序设计任务。

常用系统内部函数又称为内部函数(或内部方法)，它与任何对象无关，可以直接使用。

1. eval 函数

语法：

```
eval(string)
```

说明：

eval 函数的作用是返回字符串 string 表达式中的值。该函数接受原始字符串作为参

数，将该字符串作为代码在上下文环境中执行，并返回执行结果。

string 是要计算的字符串表达式，含有要计算的 JavaScript 表达式或要执行的语句。

2. escape 函数

语法：

```
escape(string)
```

说明：

escape 函数可以对字符串(ISO—Latin-1 字符集)进行编码，这样就可以在所有的计算机上读取该字符串。该函数不会对 ASCII 字符和数字进行编码，也不会对下面这些 ASCII 标点符号进行编码，即 * 、@、—、+、. 、/，其他所有的字符都会被转义。

string 是要被转义或编码的字符串。

3. unescape 函数

语法：

```
unescape(string)
```

说明：

unescape 函数返回的字符串是 ISO—Latin-1 字符集的字符，该函数找到形式为%XX 和%XXXX 的字符序列(X 表示十六进制的数字)，用字符\u00XX 和\uXXXX 替换这样的字符序列进行解码。

string 是要解码或反转义的字符串。

4. parseFloat 函数

语法：

```
parseFloat(string)
```

说明：

parseFloat 函数的作用是返回 string 字符串对应的实数值，只有字符串中的第一个数字会被返回，如果字符串 string 的第一个字符不能被转换为数字，那么 parseFloat 会返回 NaN。

string 是要被解析的字符串。

5. parselnt 函数

语法：

```
parselnt(string,radix)
```

说明：

parselnt 函数的作用是返回 string 字符串对应的十进制整数值，参数 radix 用于指定数字的基数，只有字符串中的第一个数字会被返回，如果字符串的第一个字符不能被转换为数

字，那么 parselnt 会返回 NaN。

string 是要被解析的字符串。

radix 表示要解析的数字的基数，该值介于 2～36 之间，如果省略该参数或其值为 0，则数字将以 10 为基数来解析；如果它以“0”开头，将以 8 为基数；如果它以“0x”或“0X”开头，将以 16 为基数；如果该参数小于 2 或者大于 36，则 parselnt( )将返回 NaN。

6. isNaN 函数

语法：

```
isNaN(string)
```

说明：

isNaN 函数的作用是判断 string 是否为数值，如果 string 是特殊的非数字值 NaN(或者能被转换为这样的值)，返回的值就是 true；如果 string 是其他值，则返回 false( )。

string 是要检测的值。

13.5.2 自定义函数

函数一般是由若干条语句构成的能够完成特定功能的代码，函数定义如下。

语法：

```
function 函数名(参数 1,参数 2,...,参数 N)
{
    函数体
}
```

说明：

function 是关键字，组成一个函数必须以 function 开始。

函数的命名必须遵守标识符命名规范，函数名用来在调用时使用。

一个函数既可以有参数，也可以没有参数，多个参数之间用逗号“,”分隔。不论有无参数，括号都不能省略，参数类型不需要给定。

函数体必须写在“{”和“}”内，“{”、“}”定义了函数的开始和结束。

在 JavaScript 中区分字母大小写，因此“function”这个词必须是全部字母小写，否则程序会出错。另外需要注意的是，必须使用大小写完全相同的函数名来调用函数。

13.5.3 用 return 返回函数的计算结果

函数可以在执行完后返回一个代表执行后的结果，当然函数也可以不返回任何值，返回函数的计算结果可以使用 return 语句来实现。

语法：

```
return 函数执行结果;
return;
```

说明：

第一段是有返回值的，第二段是没有返回值的。

有值返回的函数调用方式与无值返回的调用方式略有不同。无值返回可以通过事件触发、程序触发等方式调用；有值返回的函数类似于操作数，和表达式一样可以直接参加运算，不需要通过事件或程序来触发。

在函数体内使用不带返回值的 return 语句可以结束程序的运行，其后所有语句均不再执行。return 语句只能返回一个计算结果。return 语句后可以跟一个具体的值，也可以是一个变量，还可以是一个复杂的表达式。

13.5.4 函数变量的作用域

函数体是完成特定功能的代码段，在代码执行过程中总需要使用一些存放程序运行的中间结果的变量。变量分为局部变量和全局变量。局部变量是指在函数内部声明的变量，该变量只能在一段程序中发挥作用；而全局变量是指在函数之外声明的变量，该变量在整个 JavaScript 代码中发挥作用，全局变量的生命周期从声明开始到页面关闭时结束。

局部变量和全局变量可以重名，也就是说，即使在函数体外声明了一个变量，在函数体内还可以再声明一个同名的变量。在函数体内部，局部变量的优先级高于全局变量，即在函数体内同名的全局变量被隐藏了。

需要注意的是：专用于函数体内部的变量一定要用 var 关键字声明，否则该变量将被定义成全局变量，如果函数体外部有同名的变量，可能导致该全局变量被修改。

13.6 JavaScript 事件和对象

13.6.1 事件

1. 基本概念

1）事件(Event)

事件指的是文档或者浏览器窗口中发生的一些特定交互瞬间。用户可以通过侦听器(或者处理程序)来预定事件，以便在事件发生的时候执行相应的代码，例如 Click(单击事件)。

2）事件驱动(Event Driven)

由事件引发程序的响应，执行事先准备好的事件处理代码，这种程序运行方式称为事件驱动，如 onClick(单击事件驱动)。

3）事件处理代码(Event Handle)

在 JavaScript 中，事件处理代码通常被定义为函数的形式，在其中加入所需要的处理逻辑，并将之关联到所关注的事件源组件上。

2. 常用事件

JavaScript 的常用事件见表 13-12。

表 13-12 JavaScript 常用事件

事件名称	适用对象	意义	说明
Abort	image	中止	当图形尚未完全加载前,用户就单击了一个超链接,或单击了停止按钮
Blur	window 及所有表单子组件(text、button、select)	失去焦点	用户将他的输入焦点(闪烁光标)从窗口或表单移开
Chang	text、password、textarea、select	改变	用户改变组件的值
Click	link 及所有表单子组件(text、button、select)	单击鼠标一下	用户在上述对象上单击一下鼠标左键
DblClick	link 及所有表单子组件(text、button、select)	双击鼠标	用户在上述对象上连续双击鼠标
DrogDrop	window	拖曳	用户用鼠标左键将对象拖曳至窗口
Error	image、window	错误	加载文件或图形时发生错误
Focus	window 及所有表单子组件(text、button、select)	取得焦点	用户将输入焦点放进上述对象中
KeyDown	image、link 及所有表单子组件(text、button、select)	按下键盘上的任意键	用户按下键盘上的某个按键的一刹那
KeyPress	image、link 及所有表单子组件(text、button、select)	按下键盘上的任意键	用户按下键盘上的某个键时
KeyUp	image、link 及所有表单子组件(text、button、select)	键盘上的任意键弹起	用户按下键盘上的某个键后按键弹起来的一刹那
Load	document	加载	浏览器读入该文件时
MouseDown	document、link 及所有表单子组件(text、select)	单击鼠标	用户单击鼠标时
MouseMove	document、link 及所有表单子组件(text、select)	移动鼠标	用户移动鼠标时
MouseOver	document、link 及所有表单子组件(text、select)	鼠标进入	用户将鼠标移动到上述对象时
MouseOut	document、link 及所有表单子组件(text、select)	移开鼠标	用户将鼠标离开上述对象时
MouseUp	document、link 及所有表单子组件(text、select)	放开鼠标左键	用户将鼠标左键放开时
Move	window	移动	用户或程序移动窗口时
Reset	form	重来一次	用户单击表单中的 reset 按钮时
Resize	window	改变大小	用户调整窗口大小
Select	text、password、textarea	选择	用户选取上述对象
Submit	form	送出	用户单击表单中的 submit 按钮时
Upload	document	退出(卸载)	用户关闭或退出目前网页

3. 事件处理

1) 鼠标事件

在网页设计中,如果用鼠标对网页中的元素进行操作会触发鼠标事件。当单击鼠标左

键时会触发 Click 事件，双击鼠标时会触发 DblClick 事件，将鼠标按下后再松开时会触发 MouseUp 事件等。

例如：

```
< input type = "button" name = "click" value = "鼠标单击" onClick = "alert('你单击了我!')>
```

就是当鼠标单击页面中的按钮时触发的鼠标单击事件。

2）键盘事件

键盘事件主要有 3 个，分别是 KeyDown、KeyPress 及 KeyUp 事件，它们用来检测键盘按下、按下松开及完全松开这些动作。按键的信息被包含在事件发生时创建的对象 event 中，用 event. keyCode 可以获得按键对应的键码值。

3）表单事件

表单是 Web 应用中和用户进行交互的最常用的工具，用户注册、发表文章和评论等都需要用到表单，用户在表单中填写数据，然后将数据发送到服务器端。JavaScript 脚本所要做的主要工作就是表单验证，如验证用户是否有未填信息、输入的数据格式是否正确等，这样，在数据被提交到服务器之前数据的正确性和合法性就得到了验证并反馈给了用户，用户可以根据提示修改错误。

表单包含的元素有很多，例如文本输入框、下拉列表框、复选框、单选按钮、提交按钮等。

表单事件包括获得焦点与失去焦点事件、提交及重置事件、改变及选择事件。

4）窗口事件

窗口事件是指浏览器窗口在加载页面或卸载页面时触发的事件。在加载页面时会触发 Load 事件，在卸载页面时会触发 UnLoad 事件，这两个事件和<body>及<frameset>两个页面元素有关。

13.6.2 常用对象

建立对象的目的是将与对象有关的属性和方法封装在一起提供给程序设计人员使用，从而减轻编程人员的工作量，提高设计 Web 页面的能力。

JavaScript 的对象类型可以分为下面 4 类。

(1) JavaScript 的本地对象(native object)：本身提供的类型，例如 Math 等。这种对象无须具体定义，直接可以通过名称引用它们的属性和方法，例如 Math. Random()。

(2) JavaScript 的内置对象(built-in object)：例如 Array、String 等。这些对象独立于宿主环境，在 JavaScript 程序中由程序员定义具体对象，并可以通过对象名来使用。

(3) JavaScript 的宿主对象(host object)：被浏览器支持，目的是为了能和被浏览的文档乃至浏览器环境交互，例如 document、window 和 frames 等。

(4) JavaScript 的自定义对象：程序员根据需要自己定义的对象类型。

对象在使用之前需要声明和实例化，在创建对象时可以使用 new 关键字后跟实例化的类的名称来创建，如通过语句“var obj＝new Object();”来创建一个 Objet 对象。在创建对象后可以通过“对象名称. 属性名”的方式来访问属性，可以通过“对象名称. 方法名()”的方式来调用对象的方法。在 JavaScript 中包含了一些常用的对象，例如 Array、Boolean、Date、Math、Number、String、Object 等，这些对象常用在客户端和服务器端的 JavaScript 中，下面

对这些常用对象做简单介绍。

1. Array

数组对象用来在单独的变量名中存储一系列的值。通过声明一个数组,将相关的数据存入数组,使用循环等结构对数组中的每个元素进行操作。

1) 定义数组并直接初始化数组元素

以下两种形式都可以用来声明并且创建一个数组元素已经初始化好的数组对象,在这里,course 是课程数组对象的名字,在代码中可以通过它来访问里面的每个元素,例如:

```
var course = new Array("C++程序设计","HTML 开发基础","数据库原理","计算机网络");
var course = ["C++程序设计","HTML 开发基础","数据库原理","计算机网络"];
```

2) 先定义数组,后初始化数组元素

前面在声明数组的同时初始化了数组的元素,除了这种方式外还可以先声明并创建数组对象,然后再向数组中指定位置赋值,例如:

```
var course = new Array();          /* 先定义数组 */
courser[0] = "C++程序设计";        /* 给数组元素 1 赋值 */
courser[1] = "HTML 开发基础";      /* 给数组元素 2 赋值 */
courser[2] = "数据库原理";         /* 给数组元素 3 赋值 */
courser[3] = "计算机网络";         /* 给数组元素 4 赋值 */
```

3) 数组的长度

在定义数组时,并没有规定数组的长度,也就是没有规定这个数组可以容纳多少个元素。JavaScript 是一种弱类型的语言,对数组长度没有特别的限制,用户可以根据需要随时增加或减少。在使用过程中,可以通过数组对象的 length 属性来获得指定数组的实际长度。获取数组 course 的长度的代码如下:

```
var len = course.length;           /* len 的值为 4 */
```

4) 数组的元素

一般来说,数组中存放的都是同种类型的数据,如字符串、整数、同种类型的对象,但由于 JavaScript 是一种弱类型的语言,不检查存入数组的每个元素的类型是否一致(在强类型的语言中(如 C、C++),编译器会自动检查元素类型是否一致),也就是说数组元素可以不一致。

作为 Web 前端开发人员,在编程时应尽量保证数组中的元素的数据类型相同,这是一种良好的编程习惯。

5) 访问/修改数组元素

数组的元素可以通过下标来访问。数组的下标总是从 0 开始,直到数组长度－1 结束。

6) 使用数组对象的属性和方法

Array 对象最常用的属性是 length 属性,通过它可以获得一个数组的长度。Array 对象最常用的方法如表 13-13 所示。

表 13-13 Array 对象的方法

方　法	说　明
concat()	连接两个或更多的数组，并返回结果
join()	把数组的所有元素放入一个字符串，元素通过指定的分隔符进行分隔
reverse()	颠倒数组中元素的顺序
pop()	删除并返回数组的最后一个元素
push(新元素)	向数组的末尾添加一个或更多元素，并返回新的长度
shift()	删除并返回数组的第一个元素
unshift()	向数组的开头添加一个或更多元素，并返回新的长度
splice()	删除元素，并向数组添加新元素
slice()	从某个已有的数组返回选定的元素
sort()	对数组的元素进行排序
toString()	把数组转换为字符串，并返回结果
toLocaleString()	把数组转换为本地数组，并返回结果

2. Date

在 Web 应用中，经常需要处理与时间和日期相关的问题。JavaScript 脚本内置了核心对象 Date，该对象可以表示从毫秒到年的所有时间和日期，并提供了一系列操作时间和日期的方法。

1) 生成日期对象

Date 对象的构造函数通过可选的参数可以生成表示过去、现在和将来的 Date 对象。其构造方式有 4 种，分别如下：

```
var MyDate = new Date();
var MyDate = new Date(milliseconds);
var MyDate = new Date(string);
var MyDate = new Date(year,month,day,hours,minutes,seconds,milliseconds);
```

2) 提取日期各字段

根据定义对象的方法，可以看出日期对象包括年月日时分秒等各种信息，Date 对象提供了获得这些内容的方法，如表 13-14 所示。

表 13-14 提取日期对象每个字段的方法

方　法　名	说　明
getDate()	从 Date 对象返回一个月中的某一天(1～31)
getDay()	从 Date 对象返回一周中的某一天(0～6)
getMonth()	从 Date 对象返回月份(0～11)
getFullYear()	从 Date 对象以四位数字返回年份
getHours()	返回 Date 对象的小时(0～23)
getMinutes()	返回 Date 对象的分钟(0～59)
getSeconds()	返回 Date 对象的秒数(0～ 59)
getMilliseconds()	返回 Date 对象的毫秒(0～999)
getTime()	返回 1970 年 1 月 1 日至今的毫秒数

3）将日期转换成字符串

Date 对象提供了一些特有的方法将日期转换为字符串，而不需要开发人员编写专门的函数去实现该功能，如表 13-15 所示。

表 13-15 将日期转换成字符串的方法

方 法 名	说 明
toString()	把 Date 对象转换为字符串
toLocaleString()	根据本地时间格式把 Date 对象转换为字符串
toLocaleTimeString()	根据本地时间格式把 Date 对象的时间部分转换为字符串
toLocaleDateString()	根据本地时间格式把 Date 对象的日期部分转换为字符串

3. Math

Math 对象拥有一系列的属性和方法，能够进行比基本算术运算更复杂的运算，但 Math 对象的所有属性和方法都是静态的，并不能生成对象的实例，却能直接访问它的属性和方法。

1）使用 Math 的属性

Math 的属性如表 13-16 所示。

表 13-16 Math 的属性

属 性 名	说 明
Math. E	返回算术常量 e，即自然对数的底数(约等于 2.718)
Math. LN2	返回 2 的自然对数(约等于 0.693)
Math. LN10	返回 10 的自然对数(约等于 2.302)
Math. LOG2E	返回以 2 为底的 e 的对数(约等于 1.414)
Math. LOG10E	返回以 10 为底的 e 的对数(约等于 0.434)
Math. PI	返回圆周率(约等于 3.14159)
Math. SQRT1_2	返回 2 的平方根的倒数(约等于 0.707)
Math. SQRT2	返回 2 的平方根(约等于 1.414)

2）使用 Math 的方法

Math 的方法如表 13-17 所示。

表 13-17 Math 的方法

方 法 名	说 明
Math. abs(*x*)	返回数的绝对值
Math. sin(*x*)	返回数的正弦
Math. cos(*x*)	返回数的余弦
Math. tan(*x*)	返回角的正切
Math. acos(*x*)	返回数的反余弦值
Math. asin(*x*)	返回数的反正弦值
Math. atan(*x*)	以介于－PI/2 和 PI/2 弧度之间的数值返回 *x* 的反正切值
Math. atan2(*x*,*y*)	返回从 X 轴到点(*x*,*y*)的角度(介于－PI/2 和 PI/2 弧度之间)

续表

方法名	说明
Math.ceil(x)	对数进行上舍入，返回大于等于 x，并且与 x 接近的整数
Math.floor(x)	对数进行下舍入，返回小于等于 x，并且与 x 接近的整数
Math.round(x)	把数四舍五入为最接近的整数
Math.random()	返回 0～1 之间的随机数
Math.max(x,y)	返回 x 和 y 中的最高值
Math.min(x,y)	返回 x 和 y 中的最低值
Math.sqrt(x)	返回数的平方根
Math.exp(x)	返回 e 的指数
Math.pow(x,y)	返回 x 的 y 次幂
Math.loge(x)	返回数的自然对数(底为 e)

Math 对象提供了很多数学方法用于基本运算，这些基本运算能够满足 Web 应用程序的要求。例如在 JavaScript 脚本中，可以使用 Math 对象的 random()方法生成 0～1 的随机数。

4. Number

Number 对象对应于原始数值类型和提供数值常数的对象，JavaScript 会自动在原始数据和对象之间转换，在编程时无须考虑创建数值对象，直接使用数值变量名即可。Number 对象常用的方法主要有下面两个。

(1) toString(radix)：直接指定进制将数值转换为字符串，默认为十进制。

(2) toFixed(n)：将 Number 四舍五入为指定 n 位小数点的数字，在多余的小数位被抛弃或不足的情况下后面补 0，n 可以是 0、正整数。

5. String

String 对象是与原始字符串数据类型相对应的 JavaScript 内置对象，属于 JavaScript 核心对象之一，主要提供诸多方法实现字符串检查、抽取子串、字符串连接、字符串分割等与字符串相关的操作，可以通过以下方式生成 String 对象，例如：

```
var s1 = "hello,world";
var s2 = new String("hello,world");
```

1) 获取目标字符串长度

字符串的长度 length 是 String 对象的唯一属性，且为只读属性，它返回目标字符串(包含字符串中的空格)所包含的字符数。

2) 连接两个字符串

String 对象的 concat()方法能将作为参数传入的字符串加入到调用该方法的字符串的末尾，并将结果返回给新的字符串。

3) 字符串查找

字符串对象提供了在字符串内查找一个子串是否存在的方法，比较常用的有 indexOf()和 lastIndexOf()方法。indexOf 方法的功能是返回某个指定的字符串值在字符串中首次出现

的位置，在一个字符串中的指定位置从前向后搜索，如果没有发现，则返回－1。lastIndexOf()的功能是返回一个指定的字符串值最后出现的位置，在一个字符串中的指定位置从后向前搜索，如果没有发现，则返回－1。

4）字符串的分割

split()方法用于把一个字符串分割成字符串数组。

5）字符串的显示风格

字符串对象还提供了很多其他方法，可以修改字符串在 Web 页面中的显示风格，这些方法如表 13-18 所示。

表 13-18 字符串显示风格的方法

方法名	说明	方法名	说明
blink()	显示闪动的字符串	big()	使用大字号来显示字符
bold()	使用粗体显示字符串	small()	使用小字号来显示字符
fontcolor()	使用指定的颜色来显示字符串	strike()	使用删除线来显示字符串
fontsize()	使用指定的尺寸来显示字符串	sub()	把字符串显示为下标
italics()	使用斜体显示字符串	sup()	把字符串显示为上标

6）字符串的大小写转换

字符串对象提供了字符串中的字符大小写互相转换的方法，如表 13-19 所示。

表 13-19 字符串大小写转换的方法

方法名	说明
toLocaleLowerCase()	把字符串按照本地方式转换为小写
toLocaleUpperCase()	把字符串按照本地方式转换为大写
toLowerCase()	把字符串转换为小写
toUpperCase()	把字符串转换为大写

6. Boolean

Boolean 对象是对应原始逻辑数据类型的内置对象，它具有原始的 Boolean 值，只有 true 和 false 两个状态。在 JavaScript 脚本中，1 代表 true 状态，0 代表 false 状态。在创建 Boolean 对象时可以用以下语句：

```
var boolean1 = new Boolean(value);
var boolean2 = Boolean(value);
```

Boolean 对象主要有 3 个方法，分别是 toSource()、toString()及 valueOf()方法。toSource()方法返回表示当前 Boolean 对象实例创建代码的字符串；toString()方法返回当前 Boolean 对象实例的字符串（“true”或“false”）；valueOf()方法得到一个 Boolean 对象实例的原始 Boolean 值。

13.6.3 BOM

在实际应用中，经常使用 JavaScript 操作浏览器窗口以及窗口上的控件，从而实现用户

和页面的动态交互。因而浏览器预定义了很多内置对象,这些对象都含有相应的属性和方法,通过这些属性和方法控制浏览器窗口及其控件。

客户端浏览器这些预定义的对象统称为浏览器对象,按照某种层次组织起来的模型统称为浏览器对象模型(Browser Object Model,BOM)。浏览器对象模型定义了浏览器对象的组成和相互关系,描述了浏览器对象的层次结构,是 Web 页面中内置对象的组织形式。浏览器对象的模型如图 13-7 所示,从该图中不仅可以看到浏览器对象的组成,还可以看到不同对象的层次关系,window 对象是顶层对象,包含了 history、document、location、screen、navigator 及 frame 对象,这些对象都含有若干属性和方法,使用这些属性和方法可以操作 Web 浏览器窗口中的不同对象控制和访问 HTML 页面中的不同内容。

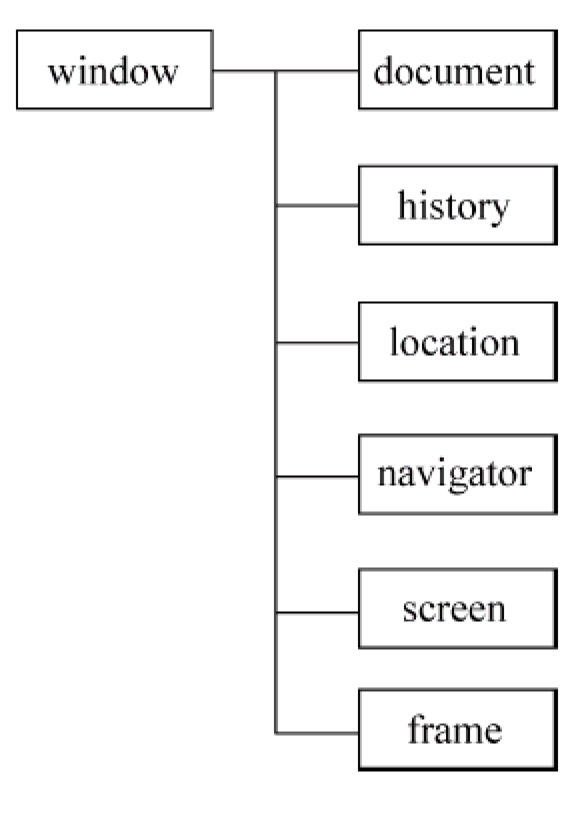


图 13-7 浏览器对象模型

1. window 对象

window 对象位于浏览器对象模型的顶层,是 document、frame、location 等对象的父类。在实际应用中,只要打开浏览器,无论是否存在页面,window 对象都将被创建。由于 window 对象是所有对象的顶层对象,所以按照对象层次访问某一个对象时不必显式地注明 window 对象。

window 对象内置了许多方法供用户操作,下面列出最常用的 window 对象的方法,如表 13-20 所示。

表 13-20 window 对象的方法

方 法 名	说 明
alert(message)	显示带有一段消息和一个确认按钮的告警框
confirm(question)	显示带有一段消息以及确认按钮和取消按钮的对话框
open(url,name,features,replace)	打开一个新的浏览器窗口或查找一个已命名的窗口
prompt("提示信息","默认值")	显示可提示用户输入的对话框
blur()	把键盘焦点从顶层窗口移开
close()	关闭浏览器窗口
focus()	把键盘焦点给予一个窗口
setInterval(code,interval)	按照指定的周期(以毫秒计)来调用函数或计算表达式
setTimeout(code,delay)	在指定的毫秒数后调用函数或计算表达式
clearInterval(intervalID)	取消由 setInterval()设置的 timeout
clearTimeout(timeoutID)	取消由 setTimeout()方法设置的 timeout

window 对象提供有 3 种用于客户与页面交互的对话框,分别是告警框、确认框和提示框。

2. navigator 对象

navigator 对象用于获取用户浏览器的相关信息。该对象是以 Netscape Navigator 命

名的，在 Navigator 和 Internet Explorer 中都得到了支持。navigator 对象包含若干属性，主要用来描述浏览器的信息，但不同浏览器所支持的 navigator 对象的属性是不同的，常用的属性如表 13-21 所示。

表 13-21 navigator 对象的属性

属 性 名	说 明
appName	返回浏览器的名称
appVersion	返回浏览器的平台和版本信息
platform	返回运行浏览器的操作系统平台
systemLanguage	返回操作系统使用的默认语言
user Agent	返回由客户机发送服务器的 user-Agent 头部的值
appCodeName	返回浏览器的代码名

另外，navigator 对象还支持一系列的方法，与属性一样，不同浏览器支持的方法也不完全相同。其常用的方法如表 13-22 所示。

表 13-22 navigator 对象的方法

方 法 名	说 明
taintEnabled()	规定浏览器是否启用数据污点(data tainting)
JavaEnabled()	规定浏览器是否启用 Java
preference()	查询或者设置用户优先级，该方法只能用在 Navigator 浏览器中
savePreference()	保存用户的优先级，该方法只能用在 Navigator 浏览器中

3. screen 对象

screen 对象用于获取用户屏幕设置的相关信息，主要包括显示尺寸和可用颜色的数量信息。表 13-23 中给出了 screen 对象的常用属性，这些属性得到了各种浏览器的普遍支持。

表 13-23 screen 对象的方法

方 法 名	说 明
availWidth	返回可用的屏幕宽度
availHeight	返回可用的屏幕高度
height	返回显示屏幕的高度
width	返回显示屏幕的宽度

在浏览器窗口打开的时候，可以通过 screen 对象的属性来获取屏幕设置的相关信息。

4. history 对象

history 对象表示窗口的浏览历史，并由 window 对象的 history 属性引用该窗口的 history 对象。history 对象是一个数组，其中的元素存储了浏览历史中的 URL，用来维护在 Web 浏览器当前会话内所有曾经打开的历史文件列表。history 对象有 3 个常用方法，如表 13-24 所示。

表 13-24　history 对象的方法及说明

方　法　名	说　　明
forward()	加载 history 列表中的下一个 URL
back()	加载 history 列表中的前一个 URL
go(number|URL)	加载 history 列表中的某个具体页面。URL 参数指定要访问的 URL，number 参数指定要访问的 URL 在 history 的 URL 列表中的位置

history 对象的这 3 个方法与浏览器软件中的【后退】和【前进】按钮的功能一致。需要注意的是，如果没有使用过【后退】按钮或跳转菜单在历史记录中移动，而且 JavaScript 没有调用 history. back()或 histroy. go()方法，那么调用 history. forward()方法不会产生任何效果，因为浏览器已经处在 URL 列表的尾部，没有可以前进访问的 URL 了。

5. location 对象

location 对象用来表示浏览器窗口中加载的当前文档的 URL，该对象的属性说明了 URL 中的各个部分。location 对象的常用属性如表 13-25 所示。

表 13-25　location 对象的属性

属　性　名	说　　明
href	设置或返回完整的 URL
host name	设置或返回 URL 中的主机名
protocol	设置或返回当前 URL 的协议
port	设置或返回当前 URL 的端口号
pathname	设置或返回当前 URL 的路径部分
host	设置或返回 URL 中的主机名和端口号的组合

通过设置 location 对象的属性可以修改对应的 URL 部分，而且一旦 location 对象的属性发生变化，相当于生成了一个新的 URL，浏览器便会尝试打开新的 URL。虽然可以通过改变 location 对象的任何属性加载新的页面，但是一般不建议这么做，正确的方法是修改 location 对象的 href 属性，将其设置为一个完整的 URL 地址，从而实现加载新页面的功能。

location 对象和 document 对象的 location 属性是不同的，document 对象的 location 属性是一个只读字符串，不具备 location 对象的任何特性，所以不能通过修改 document 对象的 location 属性实现重新加载页面的功能。

location 对象除了上面所述的属性外，还有 3 个常用的方法，用于实现对浏览器位置的控制，如表 13-26 所示。

表 13-26　location 对象的方法

方　法　名	说　　明
reload()	重新加载当前文档
assign()	加载新文档
replace()	用新的文档替换当前文档

13.6.4 DOM

1. DOM 简介

document 对象是客户端 JavaScript 最常用的对象之一，在浏览器对象模型中，它位于 window 对象的下一层级。document 对象包含一些简单的属性，提供了有关浏览器中显示文档的相关信息，例如该文档的 URL、字体颜色、修改日期等。另外，document 对象还包含一些引用数组的属性，这些属性可以代表文档中的表单、图像、链接、锚以及 applet。和其他对象一样，document 对象还定义了一系列的方法，通过这些方法可以使 JavaScript 在解析文档时动态地将 HTML 文本添加到文档中。

正是由于 document 对象特有的重要性，所以从它出现开始就在不停地扩展。遗憾的是，一开始 document 对象的扩展并没有统一的规范，不同的浏览器有不同的定义，而且彼此不兼容。为了解决不兼容带来的问题，万维网联盟(W3C)制定了一种规范，目的是创建一个通用的文档对象模型(Document Object Model，DOM)，得到所有浏览器的支持。DOM 也是一个发展中的标准，它指定了 JavaScript 等脚本语言访问和操作 HTML 或者 XML 文档各个结构的方法，随着技术的发展和需求的变化，DOM 中的对象、属性和方法也在不断变化。

DOM 的设计是以对象管理组织(OMG)的规约为基础的，因此可以用于任何编程语言。最初人们把它认为是一种让 JavaScript 在浏览器间进行移植的方法，不过 DOM 的应用已经远远超出这个范围。DOM 技术使得用户页面可以动态地变化，例如可以动态地显示或隐藏一个元素、改变元素的属性、增加一个元素等，DOM 技术使得页面的交互性大大增强。

2. DOM 节点树

DOM 是一种在加载 Web 页面时由浏览器创建的 HTML 文档模型。页面按照规则由<html>、<body>、<form>和<input>等标记组成规范文档，这些标记之间存在着一定的关系，例如<body>被<html>包含，而<form>标记又被包含在<body>内，这些页面元素好像倒垂的一棵树一样，而顶端就是<html>，我们把这种描述页面标记关系的树形结构称为 DOM 节点树(文档树)，如图 13-8 所示。

3. DOM 节点

根据 HTML DOM 规范，HTML 文档中的每个成分都是一个节点，具体的规定如下：

- 整个文档是一个文档节点。
- 每个 HTML 标记是一个元素节点。
- 包含在 HTML 元素中的文本是文本节点。
- 每一个 HTML 属性是一个属性节点。
- 注释属于注释节点。

从 DOM 约束中可以看出每一个节点对应一个对象，代表该页面上的某个元素。对象 document 实际上就是该页面上所有页面元素对象的集合，通过 document 对象可以访问页面中的所有元素。DOM 上的每个节点彼此都有等级关系，并且每个节点都包含着关于自身的大量信息。一个网页最外层的标记是<html>标记，它也是页面所有元素的根，通过

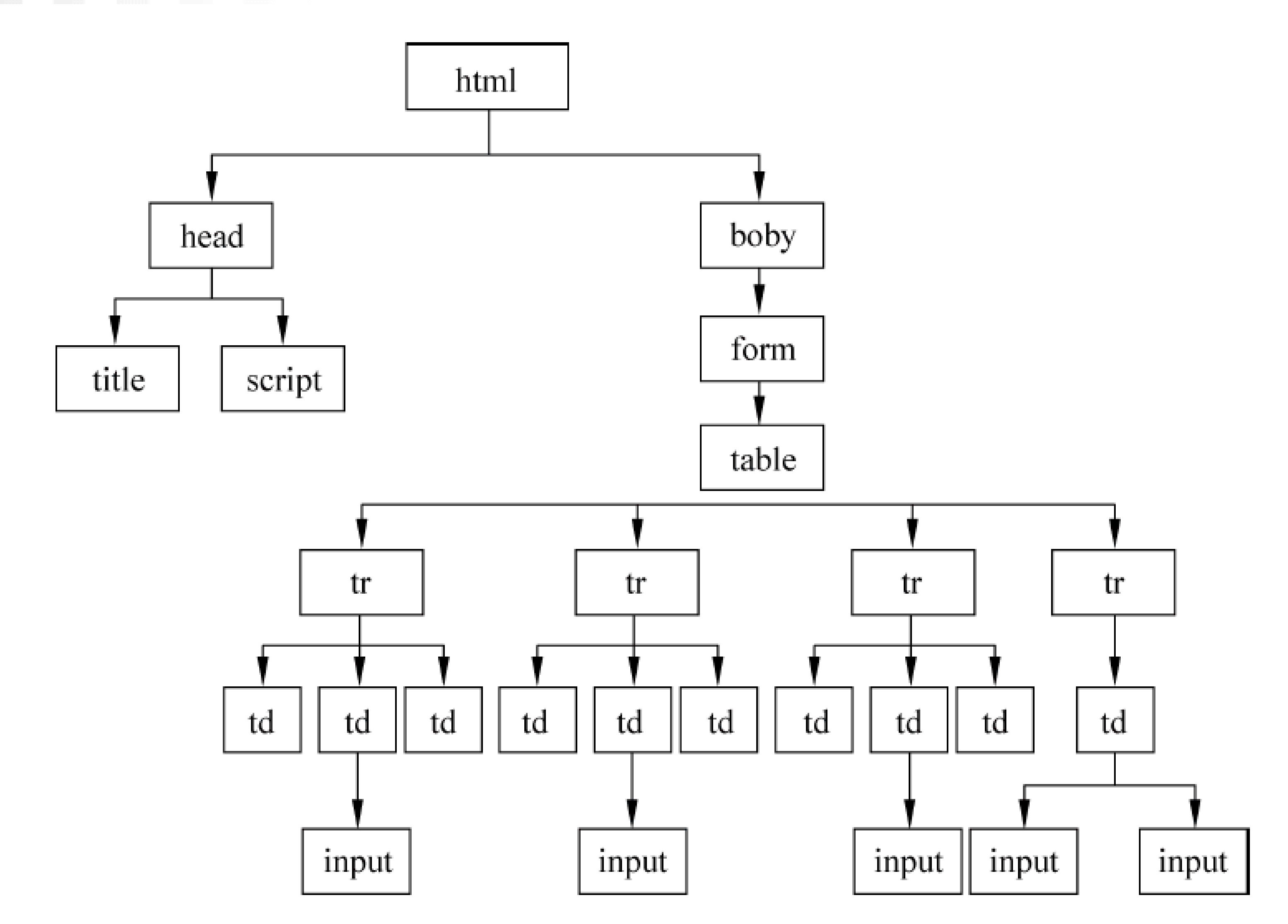


图 13-8 DOM 节点树

document 对象的 documentElement 属性可以获得，例如：

```
var root = document.documentElement;
```

由一个节点分出的下一个层次的节点称为“子节点”，而该节点称为“父节点”，即节点之间是一种父子关系。在图 13-8 中 head 节点就是 title 节点的父节点，而 title 节点是 head 节点的子节点。任何节点都可以通过数组属性 childNodes 获得自己的子节点，例如：

```
var aNodeList = root.childNodes;
```

一个节点的子节点还可以通过节点的 firstChild 和 lastChild 属性获得它的第一个和最后一个子节点。DOM 规定一个页面只有一个根节点，根节点是没有父节点的，除此之外，其他节点都可以通过 parentNodes 属性获得自己的父节点，例如：

```
var parentNode = bNode.parentNode;
```

同一父节点下位于同一层次的节点称为“兄弟节点”。在图 13-8 中 head 节点下的子节点 title 节点以及 script 节点就互为“兄弟节点”。从 DOM 树中可以看出根节点没有父节点，而最末端的节点没有子节点。不同节点对应的 HTML 元素是不同的，因此节点有不同类型。文档树中的每个节点对象都有 nodeType 属性，该属性返回节点的类型，常用的节点类型及其说明如表 13-27 所示。

表 13-27 常用的节点类型及说明

节点类型	nodeType 值	说明
Document	9	文档节点，表示当前文档
Element	1	元素节点，表示文档中的 HTML 元素
Text	3	文本节点，表示文档中的文本内容
Comment	8	注释节点，表示文档中的注释内容
Attr	2	属性节点，表示文档中 HTML 元素的属性

从表 13-27 中可以看出，如果某个节点的 nodeType 的值为 9，则说明该节点对象是一个 Document 对象，如果某个节点的 nodeType 值为 1，则说明节点对象是一个 Element 对象。不同类型的节点还可以包含其他类型的节点，相互连接在一起构成了一个完整的树形结构。对于大多数 HTML 文档来说，元素节点、文本节点和属性节点是必不可少的。

1）元素节点(Element Node)

元素节点构成了 DOM 基础。在文档结构中，<html>、<head>、<body>、<hi>、<p>和<ul>等标记都是元素节点。各种标记提供了元素的名称，如文本段落元素的名称是 p、无序列表元素的名称是 ul 等。元素可以包含其他元素，也可以被其他元素包含。图 13-8 显示了这种包含与被包含的关系，唯独 html 元素没有被其他元素包含，因为它是根元素，代表整个文档。

2）文本节点(Text Node)

元素节点只是节点树中的一种类型，如果文档完全由元素组成，那么这个文档本身将不包含任何信息，因此文档结构也就失去了存在的价值。在 HTML 文档中，文本节点包含在元素节点内，如 h1、p、li 等节点就可以包含一些文本节点。

3）属性节点(Attribute Node)

元素一般都会包含一些属性，属性的作用是对元素做出更具体的描述。例如，一般元素都有 title 属性，该属性能够对元素进行详细的描述或说明，以使用户清楚该元素的用途、作用或功能。

4. DOM 节点的访问

访问节点的方式有很多，可以通过 document 对象的方法访问节点，也可以通过元素节点的属性访问节点。具体的访问节点的方式如下：

1）通过 getElementById()方法访问节点

通过 document 对象的 getElementById()方法可以访问页面中的节点，在使用该方法时必须指定一个目标元素的 id 作为参数。

语法：

```
var s = document.getElementById("id");
```

在使用该方法时需要注意以下两点：

- id 为必选项，对应于页面元素属性 id 的属性值，类型为字符串型。在设计页面时最好给每一个需要交互的元素设定一个唯一的 id，以便查找。
- 该方法返回的是一个页面元素的引用，如果页面上出现了不同元素使用同一个 id 的情况，则该方法返回的只是第一个找到的页面元素；如果给定的 id 没有找到对应的元素，则返回 null。

2）通过 getElementsByName()方法访问节点

除了可以通过一个页面元素的 id 得到该对象的引用外，在程序中还可以通过名字访问页面元素。

语法：

```
var s = document.getElementsByName("name");
```

在使用该方法时需要注意以下两点：

- name 为必选项，对应于页面元素属性 name 的属性值，类型为字符串型。该方法在调用时返回的是一个数组，即使对应于该名字的元素只有一个。
- 如果指定名字，在页面中没有相应的元素存在，则返回一个长度为 0 的数组，在程序中可以通过判断数组的 length 属性值是否为 0 来判断是否找到了对应的元素。

3）通过 getElementsByTagName()方法访问节点

除了可以通过一个页面元素的 id 和 name 获得对应的元素外，还可以通过指定的标记名称获得页面上所有这一类型的元素。

语法：

```
var s = document.getElementsByTagName(tagname);
```

在使用该方法时需要注意以下两点：

- tagname 为必选项，对应于页面元素的类型，是字符串型的数据。该方法在调用时返回的是一个数组，即使页面中对应于该类型的元素只有一个。
- 可以通过判断数组的 length 属性值来获知页面上有多少个对应于该类型的元素。

4）通过 form 元素访问节点

form 元素是 HTML 程序提供给用户向系统输入数据的重要对象，如果要获得页面的 form 对象，除了可以通过前面的 getElementById()、getElementByName()方法获得外，还可以通过 document 对象的 forms 属性获得这个 form 对象。

13.7 综合应用

13.7.1 显示时间特效

利用 JavaScript 制作动态显示系统时间效果，页面代码如下：

```
<! DOCTYPE html PUBLIC " - //W3C//DTD XHTML 1. 0 Transitional//EN" " http://www. w3. org/TR/
xhtml1/DTD/xhtml1 - transitional. dtd">
< html xmlns = "http://www. w3. org/1999/xhtml">
< HEAD >
< TITLE >显示时间特效</TITLE >
</HEAD >
< BODY >
< script language = JavaScript >
today = new Date();
function initArray(){
this. length = initArray. arguments. length
for(var i = 0; i < this. length; i++)
this[ i + 1] = initArray. arguments[ i] }
var d = new initArray(
"星期日",
"星期一",
"星期二",
"星期三",
"星期四",
```

```
"星期五",
"星期六");
document.write(
"< font color = ##000000 style = 'font - size:9pt;font - family: 宋体'> ",
today.getYear(),"年",
today.getMonth() + 1,"月",
today.getDate(),"日",
d[today.getDay() + 1],
"</font>" );
</script>
</BODY>
</HTML>
```

显示效果如图 13-9 所示。

2015年9月6日星期日

图 13-9　动态显示系统时间效果

13.7.2　图片特效

利用 JavaScript 制作图片光感效果，代码如下：

```
<! DOCTYPE html PUBLIC " - //W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/
xhtml1/DTD/xhtml1 - transitional.dtd">
< html xmlns = "http://www.w3.org/1999/xhtml">
< HEAD >
< TITLE >图片特效</TITLE >
</HEAD >
< body >
< SCRIPT LANGUAGE = "JavaScript">
i = 0;
function f_wave()
{i = i - 4;
ShowCool.style.filter = "Wave(Freq = 1,LightStrength = 20,Phase = " + i + ")";
setTimeout("f_wave()",100);
}
window.onload = f_wave;</SCRIPT >
< IMG SRC = "01.jpg.jpg" WIDTH = "400" HEIGHT = "300" ID = ShowCool >
</BODY >
</HTML >
```

效果如图 13-10 所示。



图 13-10　图片光感效果

第14章 Ajax

本章学习目标：

✍了解 Ajax 的功能、特点。

✍了解 Ajax 的原理。

✍掌握 Ajax 中 JSON 的应用。

14.1 Ajax 概述

Ajax 是 Asynchronous JavaScript And XML（异步 JavaScript 及 XML）的缩写，2005 年由 Adaptive Path 公司的 Jesse James Garrett 首先提出。Ajax 被广泛用于大量 B/S 结构的应用中，改进了传统的 Web 应用，给浏览者一种更连续的体验。Ajax 的最大优势在于异步交互，即用户在浏览页面时可同时向服务器发送请求，甚至可以不用等待前一次请求得到完全响应便再次发送请求，这种异步请求的方式非常类似于传统的桌面应用。通过使用 Ajax 技术可以使互联网网页具有更友好的人机交互和更美观的界面。

使用 Ajax 的异步请求方式，浏览器无须频繁地重新加载新页面，服务器的响应不再是整个页面内容，而只是必须更新的部分数据。Ajax 可以减轻服务器和带宽的负担，提供更好的服务响应。使用 Ajax 的异步模式，浏览器无须重新加载整个页面就可以显示新的数据。浏览器通过 JavaScript 代码向服务器发送请求，JavaScript 代码负责解析服务器的响应数据，并把样式表加到数据上，然后在现有网页中显示出来。

Ajax 技术是 Web 2.0 的核心技术，现在已经很难找到一个没有使用 Ajax 技术的 Web 应用。

1. 为什么使用 Ajax

在 Ajax 中采用的是一系列已有的甚至是老旧的技术，把它们重新组织，扩展原有的概念，让用户可以应对客户端程序所面对的复杂问题。

早期的 Web 应用程序主要基于 HTTP 协议，主要内容是文档和超链接。用户通过单击超链接进入一个个文档，实现系统的功能。每一次单击都会引起一次页面的刷新，一次页面的刷新就意味着一次客户端与服务器端的通信，网络通信是一件代价很高昂的事情，很可能会产生网络延迟。大量的远程调用会把系统拖慢，使用户觉察到延迟，导致用户体验很差。

Ajax 技术采用异步交互的方式解决网络延迟造成的用户界面阻塞问题，从而达到提高

用户体验的目的。Ajax 技术的最大特点是：

（1）页面局部更新，不会导致整个页面完全刷新。

（2）很好的用户体验。

2. Ajax 的应用

当前很多 Web 应用都采用了 Ajax 技术，例如 Google 公司的 Gmail、Google Maps 和 Microsoft 公司的 Hotmail 等，下面通过 Google Maps 应用来看一下 Ajax 的应用。

Google Maps 结合了地图浏览和搜索引擎的功能，这个地图可以自由地通过文本来查询并且精确地细化到街道地址和生活设施。Google Maps 的地图支持鼠标的拖动、放大和缩小。地图是由很多小块的图片拼接而成的，如果用户滚动得很远，会出现一些空白区域，这些区域的图片会延时加载。这个延时很明显，可以观察到起初它们是一些空白区域，当它们被加载时会一块一块地显示出来。值得注意的是，这种数据的加载都是在地图的某些区域进行的，而不是整个页面在加载。

当使用鼠标单击地图上的地名时，地图上出现该地点的详细介绍，注意到这种操作并不会引起页面的刷新，如图 14-1 所示。



图 14-1　Google Maps 操作效果

14.2　Ajax 原理

Ajax 基本上把 JavaScript 技术和 XMLHttpRequest 对象放在 Web 表单和服务器之间。当用户填写表单时，数据发送给一些 JavaScript 代码而不是直接发送给服务器。

14.2.1　Ajax 的组成部分

Ajax 不只是一种技术，实际上它是由几种蓬勃发展的技术以新的强大方式组合而成，Ajax 包含以下组成部分。

(1) HTML 和 DOM：浏览器将 HTML 页面组织成 DOM 对象，Ajax 程序可以使用 DOM 有效地重绘页面中的部分内容。

(2) CSS：Ajax 程序可以使用 CSS 修改用户界面的外观。

(3) XML 和 JSON：用于在客户端和服务器之间传递数据。XML 是标准的数据表示格式，JSON(JavaScript Object Notation)在客户端有着优良的性能，可以减少网络流量。

(4) XMLHttpRequest：Ajax 的核心部分，允许开发者在 JavaScript 中以异步方式向服务器发出 HTTP 请求并得到响应。

(5) JavaScript：JavaScript 将以上技术结合在一起。

Ajax 的各部分组织结构如图 14-2 所示。

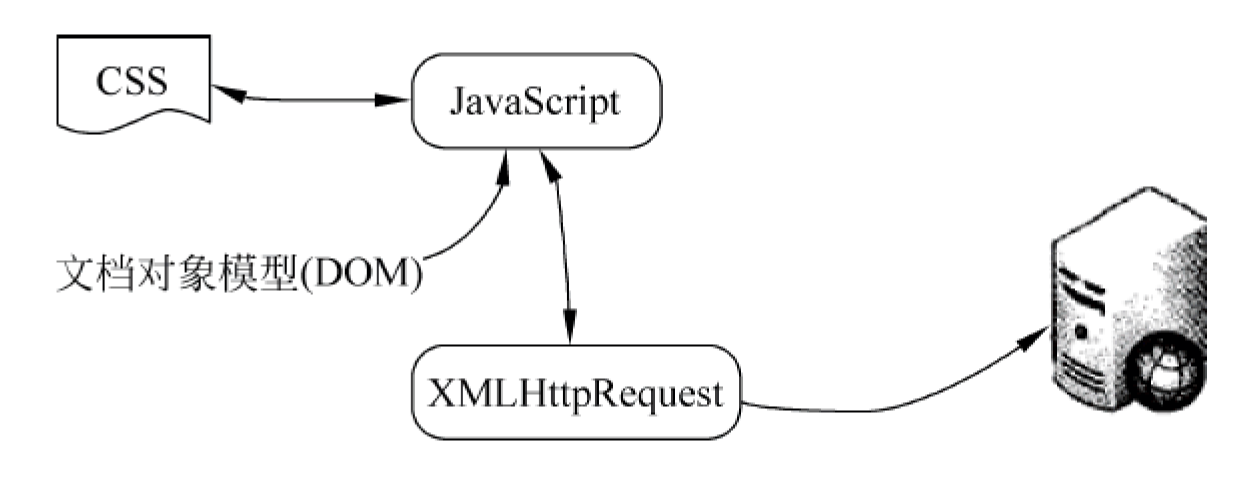


图 14-2　Ajax 组织结构图

图 14-2 所示为 Ajax 各组成元素在浏览器和 Web 服务器进行交互的过程中所起作用的描述。页面被浏览器组织成 DOM 形式，CSS 影响页面的样式，JavaScript 控制 DOM 和 CSS，通过 XMLHttpRequest 对象与 Web 服务器进行 Ajax 交互，数据格式可以是 XML 或 JSON 格式，甚至是普通文本格式。

14.2.2　传统 Web 应用和 Ajax 应用

传统 Web 应用和 Ajax 应用的主要区别在于传统 Web 应用采用同步方式与服务器端交流内容，而 Ajax 应用通过异步方式与服务器端交流数据。

传统的 Web 应用模式采用的是同步方式与服务器端交流，用户的体验是割裂的，即单击→等待→看到新的页面→再单击→再等待。在采用了 Ajax 技术之后，采用异步方式与服务器端交流数据，大部分的计算工作都是在用户不察觉的情况下交由服务器去完成，不影响用户界面的响应，另外没有空白的屏幕，没有让人心烦的旋转地球，没有虚假的下载百分比进度条，用户体验相对于传统 Web 应用有很大的提高。

传统的 Web 应用与 Ajax 应用的不同还体现在两者与服务器端交流的内容上。传统 Web 应用向服务器端发出 HTTP 请求，从服务器端得到 HTML 页面作为响应。Ajax 应用也是向服务器端发出 HTTP 请求，但从服务器端得到的响应只是要改变数据，数据的格式一般是 XML 或者 JSON。

14.3　应用 Ajax 的步骤

下面通过一个实例来展示 Ajax 各部分的使用。在 Ajax 的各种技术组成中，最重要的是 XMLHttpRequest 对象，通过该对象实现与服务器端的异步交互。开发一个 Ajax 应用

的一般步骤如下：

- 初始化 XMLHttpRequest 对象；
- 设置 XMLHttpRequest 对象的 onreadstatechange 属性，指定服务器返回响应数据时要调用的回调函数；
- 调用 XMLHttpRequest 对象的 open 方法，创建 HTTP 请求；
- 处理服务器返回的信息。

1. 初始化 XMLHttpRequest 对象

Ajax 的核心是 JavaScript 对象 XMLHttpRequest。该对象不是浏览器中 DOM 的标准扩展，该对象在 Internert Explorer 5 中首次引入，它提供了异步发送请求的能力。后来，其他浏览器厂商也实现了类似的功能对象，使得 XMLHttpRequest 对象成为事实上的行业标准。使用 XMLHttpRequest 可以通过 JavaScript 向服务器发送请求，并能够处理服务器响应，避免阻塞用户动作。通过使用 XMLHttpRequest 对象，浏览器通过客户端脚本与服务器交换数据，Web 页面无须频繁重新加载，Web 页面的内容也由客户端脚本动态更新。

为了让 JavaScript 可以向服务器发送 HTTP 请求，必须使用 XMLHttpRequest 对象。在使用前，要先将 XMLHttpRequest 对象初始化、实例化。各个浏览器对 XMLHttpRequest 对象的创建方式不尽相同。

在 Mozilla Firefox 浏览器中通过以下代码完成初始化：

```
var xmlhttp = new XMLHttpRequest();
```

Microsoft 公司的 Internet Explorer 中以 ActiveXObject 控件的形式提供，初始化代码如下：

```
var xmlhttp = new ActiveXObject("Microsoft. XMLHTTP");
```

下面的代码以一种更通用（能实现跨浏览器访问）的方式创建 XMLHttpRequest 对象：

```
//得到 xmlHttp 对象
function loadXmlHttp(){
    if(window.XMLHttpRequest){                          //IE 7、Mozilla、Safari
        xmlHttp = new XMLHttpRequest();
    }
    else if(window.ActiveXObject){
    try(
        xmlHttp = new ActiveXObject("Microsoft. XMLHTTP");    //IE 5. x,6
    catch(e){}
    }
}
```

2. 指定响应处理函数

响应处理函数（回调函数）即当服务器返回 HTTP 请求响应信息时客户端的处理方式。为 Ajax 指定响应处理函数有点类似桌面应用程序的监听事件和事件处理函数，只不过在 Ajax 应用程序中触发事件并调用事件处理函数与响应处理函数被回调执行是在不同的时

间段发生的，不在一个线程中。XMLHttpRequest 对象能够监听 HTTP 请求的状态，当状态变化时，调用回调函数执行响应处理函数。

每当 readyState 改变时就会触发 onreadystatechange 事件，readyState 属性存有 XMLHttpRequest 的状态信息。下面是 XMLHttpRequest 对象的 3 个重要的属性，如表 14-1 所示。

表 14-1 readyState 状态

属　　性	描　　述
onreadystatechange	存储函数(或函数名)，每当 readyState 属性改变时就会调用该函数
readyState	存有 XMLHttpRequest 的状态，从 0 到 4 发生变化 0：请求未初始化 1：服务器连接已建立 2：请求已接收 3：请求处理中 4：请求已完成，且响应已就绪
status	200："OK" 404：未找到页面

在 JavaScript 中，只要将相应的 JavaScript 处理函数名称给 XMLHttpRequest 对象的 onreadystatechange 属性就可以了，代码如下：

```
xmlHttp.onreadystatechangeo = onCallback;
```

其中，onCallback 是一个 JavaScript 函数。

3. 发出 HTTP 请求

在指定响应处理函数后，就可以向服务器发出 HTTP 请求了，这一步需要调用 XMLHttpRequest 对象的 open 和 send 方法：

```
xmlHttp.open("GET",url,true);
xmlHttp.send(null);
```

open 方法首先创建一个针对目标 URL、采用相应 HTTP 请求方式的 HTTP 请求。

open 方法的第一个参数是 HTTP 请求的方式，为 GET、POST 或者是 HEAD。

open 方法的第二个参数是目标 URL，这个 URL 可以是任何的 URL，包括需要服务器解释执行的页面或者是静态页面。

open 方法的第三个参数指定 HTTP 请求是否异步执行，即在等待服务器返回信息的时间内是继续执行下面的代码，还是等待服务器返回响应数据再执行下面的代码。如果为 true，则 HTTP 请求采用异步方式，即在服务器接收并响应 HTTP 请求期间，其下方代码继续执行，当服务器返回响应数据时，回调函数将被调用执行；如果为 false，则下面的代码不会执行，直到服务器返回信息。默认为 true。

open 方法在调用完毕要调用 send 方法，将 HTTP 请求发送给服务器。在调用 open 方法之后、send 方法之前，必须调用 XMLHttpRequest 对象的 setRequestHeader 方法，修改 HTTP 请求头部中的 MIME 类别。代码如下：

```
xmlHttp.setRequestHeader('Content-type\','application/x-www-form-urlencoded');
```

一般会把以上代码封装到一个函数中，以方便使用。代码如下：

```
//向服务器发出异步请求
function sendRequest(url){
    if(xmlHttp){
    xmlHttp.open("GET",url,true);
    xmlHttp.onreadystatechange = onCallback;                    //设定状态改变时触发的回调函数
    xmlHttp. setRequestHeader('Content-type\','application/x-www-form-urlencoded');
    xmlHttp.send(null);
    }
}
```

4. 处理服务器返回的信息

在发出 HTTP 请求后，XMLHttpRequest 对象等待服务器端的响应。当服务器返回信息时，回调函数会被调用。在回调函数中，首先要检查 XMLHttpRequest 对象的 readyState 值，判断请求的当前状态。当 readyState 的值为 4 时，代表 HTTP 请求发送成功并且服务器已经传回所有数据，可以开始处理信息并更新页面了。判断代码如下：

```
if(xmlHttp.readyState == 4){
//请求成功返回
}else{
//信息没有返回
}
```

在服务器返回信息后，还需要判断返回的 HTTP 请求的状态码，确定返回的数据没有错误，状态码可以参见表 14-1。其中，200 代表数据返回成功。其判断过程如下：

```
if(xmlHttp.status == 200){
//页面正常，可以开始处理信息
}else{
//页面有问题
}
```

XMLHttpRequest 对成功返回的信息有两种处理方式，即 responseText 和 responseXML。其中，responseText 将返回的信息当成字符串使用，responseXML 将返回的信息当成 XML 文档使用。使用 responseXML 处理响应数据必须保证服务器返回的响应数据是 XML 文档格式。

14.4 JSON

在 Ajax 应用程序中发送和接收信息时可以选择以纯文本和 XML 作为数据格式。对于纯文本方式，经常采用 JSON。JSON 是 JavaScript 对象表示法(JavaScript Object Notation)的简称，JSON 是一种数据描述格式，用于对数据进行编码。下面的代码就是 JSON 数据形式的一个例子：

```
{"people":[
{"firstName":"Brett","lastName":"McLanghlin","email":"brett@newInsstance.com"},
{"firstName":"Jaso","lastName":"Hunter","email":"jason@servlets.com"},
{"firstName":"Elliotte","lastName":"Harold","email":"elharo@macfaq.com"}
]}
```

JSON 使用 JavaScript 语法来描述数据对象，但是 JSON 仍然独立于语言和平台。JSON 解析器和 JSON 库支持许多不同的编程语言。JSON 文本格式在语法上与创建 JavaScript 对象的代码相同。由于这种相似性，无须解析器，JavaScript 程序能够使用内建的 eval()函数，用 JSON 数据生成原生的 JavaScript 对象。

用户可以将 JSON 和 XML 进行比较，以更深入地认识 JSON。类似于 XML，JSON 是纯文本，具有“自我描述性”(人类可读)，具有层级结构(值中存在值)，可通过 JavaScript 进行解析，数据可使用 Ajax 进行传输。

与 XML 相比，不同之处是 JSON 没有结束标签，书写字符串更短、读/写速度更快，能够使用内建的 JavaScript eval()方法进行解析，使用数组，不使用保留字。

在 Ajax 应用中，JSON 比 XML 更快、更易使用。

14.4.1 JSON 语法

JSON 语法是 JavaScript 对象表示语法的子集，它的主要要求如下：

(1) 数据在名称/值对中；

(2) 数据由逗号分隔；

(3) 花括号保存对象；

(4) 方括号保存数组。

JSON 数据的书写格式是名称/值对。名称/值对包括字段名称(在双引号中)，后面写一个冒号，然后是值，例如：

```
"firstName":"John"
```

这很容易理解，等价于下面这条 JavaScript 语句：

```
firstName = "John"
```

JSON 值可以是：

(1) 数字(整数或浮点数)；

(2) 字符串(在双引号中)；

(3) 逻辑值(true 或 false)；

(4) 数组(在方括号中)；

(5) 对象(在花括号中)；

(6) null。

14.4.2 JSON 对象

按照最简单的形式，可以用下面的 JSON 表示名称/值对：

```
{"firstName":"Brett"}
```

这个示例非常基本，而且实际上比等效的纯文本名称/值占用更多的空间：

```
firstName = "Brett"
```

但是，当将多个名称/值对串在一起时，JSON 就体现出它的价值了。首先可以创建包含多个名称/值对的记录，例如：

```
{"firstName":"Brett","lastName":"McLanghlin","email":"brett@newInsstance.com"}
```

从语法方面来看，这与名称/值对相比并没有很大的优势，但是在这种情况下 JSON 更容易使用，而且可读性更好。例如，它明确地表示以上 3 个值都是同一记录的一部分，花括号使这些值有了某种联系。

当需要表示一组值时，JSON 不仅能够提高可读性，而且可以减少复杂性。例如，假设希望表示一个人名列表，在 XML 中需要许多开始标记和结束标记；如果使用典型的名称/值对（就像在本书前面看到的那种名称/值对），那么必须建立一种专有的数据格式，或者将键名称修改为 personl-firstName 的形式；如果使用 JSON，则只需将多个带花括号的记录分组在一起。

14.4.3 在 JavaScript 中使用 JSON

读者掌握了 JSON 的格式之后，在 JavaScript 中使用它就很简单了。JSON 是 JavaScript 原生格式，这意味着在 JavaScript 中处理 JSON 数据不需要任何特殊的 API 或工具包。

1. 将 JSON 数据赋给变量

例如可以创建一个新的 JavaScript 变量，然后将 JSON 格式的数据字符串直接赋给它，代码如下：

```
    var people =
    {"programmers:[
{"firstName":"Brett","lastName":"McLaughlin","email":"brett@newlnstance.com"},
{"firstName":"Jason","lastName":"fIunter","email":"jason@servlets.com"},
{"firstName":"Elliotte","lastName":"Harold","email":elharo@macfaq.com}
],
    "authors":[
        {"firstName":"Isaac","lastName":"Asimov","genre":"science fiction"},
        {"firstName":"Tad","lastName":"Williams","genre":"fantasy"},
        {"firstName":"Frank","lastName":"peretti","genre":"christian fiction"}
        ],
    "musicians":[
    {"firstName":"Eric","lastName":"Clapton"," instrument":"guitar"},
    {"firstName":"Sergei","lastName":"Rachmaninoff","instrument":"piano"}
    ]
}
```

这非常简单：现在 people 包含前面看到的 JSON 格式的数据，但是这还不够，因为访问数据的方式似乎还不明显。

2. 访问数据

尽管看起来不明显，但是上面的长字符串实际上只是一个数组，将这个数组放进 JavaScript 变量之后就可以很轻松地访问它。实际上，只需用点号表示法来表示数组元素。所以，要想访问 programmers 列表中第一个条目的姓氏，只需在 JavaScript 中使用下面的代码：

```
people.programmers[0].lastName;
```

注意：数组索引是从 0 开始的，所以这行代码首先访问 people 变量中的数据，然后移动到被称为 programmers 的条目，再移动到第一个记录([0])，最后访问 lastName 键的值，结果是字符串值"McLaughlin"。

3. 修改 JSON 数据

正如可以用点号和括号访问数据一样，用户也可以按照同样的方式轻松地修改数据：

```
people.musicans[1].lastName = "Rachmaninov";
```

在将字符串转换为 JavaScript 对象之后，就可以像这样修改变量中的数据。

4. 转换回字符串

当然，如果不能轻松地将对象转换回本文提到的文本格式，那么所有数据修改都没有太大的意义。在 JavaScript 中这种转换也很简单：

```
String newJSONtext = people.toJSONString();
```

这样就行了。现在获得了一个可以在任何地方使用的文本字符串。例如，可以将它用作 Ajax 应用程序中的请求字符串。

更重要的是，可以将任何 JavaScript 对象转换为 JSON 文本，并非只能处理原来用 JSON 字符串赋值的变量。为了对名为 myObject 的对象进行转换，只需执行相同形式的命令：

```
myObjectlnJSON == myObject.toJSONString();
```

这就是 JSON 与本书讨论的其他数据格式之间的最大差异。如果使用 JSON，只需调用一个简单的函数就可以获得经过格式化的数据，就可以直接使用了。对于其他数据格式，需要在原始数据和格式化数据之间进行转换。即使使用 Document Object Model 这样的 API（提供了将自己的数据结构转换为文本的函数），也需要学习这个 API，并使用 API 的对象，而不是使用原生的 JavaScript 对象和语法。

最终结论是，如果要处理大量的 JavaScript 对象，那么 JSON 几乎可以肯定是一个好的选择，这样可以轻松地将数据转换为可以在请求中发送给服务器端程序的格式。

14.5 综合应用

利用 Ajax 的异步传输制作网页文字的显示。

(1) aj.js 文件的代码如下：

```
<!--

var xmlHttp;
function createAjax(){
    //创建 xmlHttp 对象
    xmlHttp = window.ActiveXObject?new ActiveXObject('Microsoft.XMLHTTP'):new XMLHttpRequest();
    //如果浏览器不支持
    if(!xmlHttp){
        alert('此浏览器不支持 xmlHttpRequest 对象');
    }
}
    function changeImg(url,par){
    //alert('dddd');
    createAjax();
    //打开连接
    xmlHttp.open('get',url,true);
    //准备就绪
    xmlHttp.onReadyStateChange = function(){
        if(xmlHttp.readyState == 4){
            if(xmlHttp.status == 200){
                //xmlHttp.responseText 存放返回文本
                var getText = xmlHttp.responseText;
                //清除缓存
                xmlHttp = null;
                var img = "< img src = " + getText + ">";
                //document.getElementById(par).src = getText; 如果浏览器不支持用下面的形式
                document.getElementById(par).innerHTML = img;
                //在 login.php 文件< span id = "img">< img src = "../data/imgCode.php" height =
"18px"></span>
            }
        }
    }
    //发送请求
    xmlHttp.send(null);
}
function doAjax(url,par1,par2){
    //alert('ddd');
    createAjax();
    //1) 读取表单元素中的值< input type = 'text' id = ''>
```

```
        var getTestValue = document.getElementById(par1).value;

        //2) 读取 div 标签中的值
        //var getTestValue = document.getElementById(par1).innerHTML;
        //alert(getTestValue);

        //打开请求
        xmlHttp.open('get',url + '?id = ' + getTestValue,true);
        //准备执行,xmlHttp 需要的是一个函数,而不是返回值
        xmlHttp.onReadyStateChange = function(){
            if(xmlHttp.readyState == 1){
                document.getElementById(par2).innerHTML = '数据加载中-------';
            }
            if(xmlHttp.readyState == 4){
                if(xmlHttp.status == 200){
                    var getText = xmlHttp.responseText;
                    //清除缓存
                    xmlHttp = null;

                    //1) 适用于表单元素<input type = 'text' id = ''>
                    document.getElementById(par2).value = getText;

                    //2) 适用于<div id = ''></div>
                    //document.getElementById(par2).innerHTML = getText;

                }
            }
        }
        //发送请求
        xmlHttp.send(null);
    }

    -->
```

(2) 网页文件 testajax.html 的代码如下：

```
<html>
<head>
<script language = 'javascript' src = 'aj.js'></script>

</head>
<body>
<! -- ajax 异步传输文本实例 -->
<form>

    输入文本: <input type = 'text' name = 'test2' id = 't1' onKeyUp = "javascript:doAjax
('testajax.php','t1','t2');">
    <br />
```

```
    显示文本：< input type = 'text' name = 'test1' id = 't2'>
</form>

</body>

</html>
```

(3) 本例中的 Ajax 用 PHP 服务器配置，故存在一个 PHP 文件，testajax. php 的代码如下：

```
<?php
$ txt = $ _GET['id'];
echo $ txt;
?>
```

第15章 jQuery

本章学习目标：

✍了解 jQuery 的功能及应用。

✍掌握 jQuery 的选择器和函数。

✍掌握 jQuery UI 的使用。

15.1 jQuery 概述

jQuery 由美国人 John Resig 创建，至今已吸引了来自世界各地的众多 JavaScript 高手加入其 team，包括来自德国的 Jörn Zaefferer，罗马尼亚的 Stefan Petre 等。jQuery 是继 Prototype 之后又一个优秀的 JavaScript 框架。其宗旨是“Write Less，Do More”，即“写更少的代码，做更多的事情”。

15.1.1 jQuery 的起源

jQuery 是一个轻量级的 JavaScript 库，JavaScript 语言是在 Web 页面上使用的客户端脚本语言，使用 jQuery 可以帮助用户迅速地完成各种脚本功能，并且实现的效果都是跨浏览器兼容的。jQuery 的主要设计目标是“Write Less，Do More”，即“写更少的代码，做更多的事情”，从而改变编写 JavaScript 脚本代码的方式。

jQuery 是由 John Resig 于 2006 年 1 月推出的开源 JavaScript 项目，已经从当年的 1.0 版本发展到 2009 年的 1.3.X 版本，下面简单回顾一下 jQuery 的发展历史。

jQuery 1.0 于 2006 年 8 月发布，作为第一个发布版本已经包含了 CSS 选择器、事件处理和 Ajax 接口函数。

jQuery 1.1 于 2007 年 1 月发布，对 API 做了大量的整理，将不常用的函数进行合并或删减。

jQuery 1.1.3 于 2007 年 7 月发布，大幅度改善了选择器的性能，性能已经可以和 Prototype、MooTools、Dojo 等相媲美。

jQuery 1.2 于 2007 年 12 月发布，取消了 XPath 选择器，因为 CSS 选择器已经足够强大，对函数的易用性和插件的开发都有所改进。

jQuery UI 于 2007 年 12 月发布，jQuery UI 的发布为实现丰富的用户界面和用户体验提供了强有力的基础。

jQuery 1.2.6 于 2008 年 5 月发布，主要的更新在于性能提升，并且整合了 DimensionsPlugin 插件。

jQuery 1.3 于 2009 年 1 月发布，它更新了 Sizzle 选择器引擎，提高了很多函数方法的性能，一举让 jQuery 成为最快的脚本类库，此外还添加了 live 等事件委托函数。

jQuery 1.4 在 2010 年 1 月 14 日，jQuery 四岁生日时发布，该版本显著提高了最常用的 jQuery 方法的性能，并且修复了非常多的 bug，jQuery 开发团队在开发 jQuery 1.4 时大幅度增加了测试用例。jQuery 在所有主流浏览器中全部通过，1.4 版本还完全更换了 jQuery 的在线文档手册。同时启用了新的 api.jQuery.com 作为手册的地址，此手册对函数的组织和分类更加系统化。

通过自身不断地完善和更新，jQuery 的应用已经越来越广泛，到目前为止它是使用最广泛的 JavaScript 类库之一，被微软和诺基亚等各大公司采用，只要是需要使用 JavaScript 的地方，jQuery 就能发挥重要作用。

15.1.2 jQuery 的功能

1. 获取网页元素

HTML DOM(Document Object Model，文档对象模型)定义了访问和处理 HTML 文档的标准方法，它将 HTML 文档呈现为带有元素、属性和文本的树结构(节点树)。虽然可以通过 JavaScript 重构整个 HTML 文档，可以添加、移除、改变或重排网页中的元素，但是如果不使用 JavaScript 库，而直接使用原生的 JavaScript 脚本来访问 HTML 文档中的某个元素，则必须编写冗长的代码，jQuery 提供了一套高效、可靠的选择器，将这些选择器与工厂函数 $()结合使用，就可以方便、快捷地从文档中选择所需的元素。

2. 设置网页外观

HTML 标记用于定义文档内容，同时由浏览器完成文档布局。虽然可以使用 CSS 定义如何显示 HTML 元素，并实现内容与表现形式的分离，不过一旦遇到了不同浏览器不完全支持相同标准的情况，仅仅使用 CSS 往往无法达到预期目的。jQuery 提供了跨浏览器的解决方案，当访问者使用当前流行的大多数浏览器时，用 jQuery 设置的网页都会保持相同的外观。

3. 更改网页内容

使用 jQuery 提供的 API 方法只需要编写少量代码就可以很轻松地修改 HTML 文档的内容，例如动态改变段落、div 元素或表格单元格包含的文本、插入或翻转网页上的图片等。

4. 响应用户操作

jQuery 提供了可靠的事件处理机制，可以在各种浏览器中捕获各种各样的网页事件，并通过代码对用户的操作做出适当响应，从而为网页添加动态性和交互性。

5. 添加动画效果

jQuery 不仅内置了一些淡入、擦除之类的效果，还提供了制作新效果的工具包，开发者可以使用这些功能为网页添加动画效果，在实现交互行为的同时提供视觉上的美观，使站点让访问者感到耳目一新。

6. 实现 Ajax 交互

Ajax 的全称是 Asynchronous JavaScript and XML，意思是异步的 JavaScript 和 XML，Ajax 是 Web 2.0 的核心技术。jQuery 对 Ajax 提供了很好的支持，借助于 jQuery，它允许以异步通信方式向服务器发出 HTTP 请求并从服务器获取响应信息，客户端可以在任何时候与服务器进行通信，而无须刷新整个页面。

15.1.3 jQuery 的应用

下面结合一个典型的例子来说明 jQuery 在 Ajax 开发中的应用。在这个例子中将模拟注册网站用户时的新用户名检测过程，分别通过原生的 JavaScript 和 jQuery 来实现 Ajax 请求，读者可以从两者的比较中对 jQuery 获得一个初步体验。

1. 配置开发环境

jQuery 只是一个用于网页客户端脚本编程的 JavaScript 库，不必进行安装和调试工作。但是，为了使用 jQuery 实现 Ajax 交互，需要架设服务器端环境。另外，为了便于编写和调试脚本代码，还需要选择一种适当的开发工具。在本书中选择 Tomcat 作为应用程序服务器，并选择 EditPlus 作为开发工具。

2. 创建 HTML 页面

用 EditPlus 创建一个网页，在该页中插入一个表单并在其中添加标签、文本框、密码框和提交按钮。源代码如下：

```
<!DOCTYPE HTML PUBLIC" - //W3C XHTML 1.0 Transitional//EN" "http://www.w3.org /TR/xhtml1/DTD/
xhtml1 - transitional.dtd">
<html xmlns = "http://www.w3.org/1999/xhtml">
<head>
<title> user register </title>
<meta http - equiv = "Content - Type" content = "text/html;charset = utf - 8"/>
<style>
    body{font - sixe:12px}
    input{width:120px}
</style>
</head>
<body>
    <form method = "post" action = "" id = "frm_register" name = "frm_register">
        <table border = "1" width = "180px">
            <tr>
                <td><lable for = "username"> username:</label></td>
```

```
                <td>
                    <input type="text" id="username" name="username"/>
                    <span id="message_a"></span>
                </td>
            </tr>
            <tr>
                <td><lable for="password">password:</label></td>
                <td>
                    <input type=" password " id=" password " name=" password "/>
                    <span id="message_b"></span>
                </td>
            </tr>
            <tr>
                <td colspan="2" align="right">
                <input type="submit" id="cmd_regisster" value="register"/>
                </td>
            </tr>
        </table>
    </form>
</body>
</html>
```

此网页是一个基于 XHTML 1.0 Transitional 标准的 HTML 文档，其中文档类型声明(DOCTYPE)和主要 HTML 标记是由 EditPlus 自动生成的，设计时在页面中添加了以下内容：

在网页首部使用 style 标记定义了一个 CSS 样式表，其中包含两条规则，一个用于设置 body 各元素的字号，另一个用于设置输入(input)元素的宽度。

在正文部分添加了一个表单并命名为 frm_register，即以 frm_register 作为 id 和 name 属性值，在客户端脚本中可借此访问表单，还将该表单的 method 属性设置为 post，指定使用 post 方法在正文中嵌入表单数据。

在表单 frm_register 中插入一个 3 行 2 列的表格，用于安排以下网页元素的布局：

1）两个 label 元素

这些 label 可以为 input 元素设置标注，而不会向用户呈现任何特殊效果，不过这里将 label 的 for 属性分别设置为 input 元素的 id，可改进鼠标用户的操作条件，当在 label 内单击文本时，浏览器就会自动将焦点转到和标签相关的表单域上。

2）3 个 input 元素

将这些元素的 type 属性分别设为 text、password 和 submit，使它们分别充当单行文本框、密码框和提交按钮。在客户端脚本中，可通过 id 和 name 访问表单域，在服务器 Java Servlet 脚本中，则可通过 HttpRequest 对象相应表单域的 name 来获取用户通过表单提交的值。

3）两个 span 元素

这些元素标记分别位于表单域旁边，用于在文档中创建一个嵌入文本容器。它们的 id 分别为 message_a 和 message_b，虽然在加载网页时不可见，但可以通过客户端脚本设置其 innerHTML 属性，以指定其开始标记和结束标记之间的 HTML 内容，也可以通过 style 属

性设置其外观。

3. 编写 jQuery 代码

当在注册页上输入用户名和密码并单击【提交】按钮时就会将表单数据发送到服务器端进行处理。如果检测到所提交的用户已被注册,则应向用户显示相应的提示信息,并让其选择新的用户名进行尝试,这是传统的用户注册处理方法。

更好的解决方案是当在文本框中输入用户名并离开此文本框时即此用户名是否可用的信息,而不必等到提交表单之后。这种解决方案需要借助于 Ajax 技术才能实现,即通过客户端 JavaScript 脚本以异步方式向服务器传递数据并获取相应的响应。

下面说明如何使用 jQuery 实现 Ajax 交互。打开 user_reglster. htm,在网页首部导入 jQuery,代码如下:

```
<script type="text/JavaScript" src="jQuery-1.8.3.min. js"></script>
```

然后使用 jQuery 编写 JavaScript 客户端脚本。源代码如下:

```
<script type="text/JavaScript">
$(document).ready(function){          //在 DOM 加载就绪时绑定一个匿名函数
$("#username").blur(function){        //设置 username 的文本框的 blur 事件处理程序
if($(this).val()=''){                 //若用户名为空
$("#message_a").html("用户名不能为空!").css("color","red");  //设置 span 的内容和颜色
$(this).focus();
return;
}
//以异步方式向服务器发送数据并将 HTTP 响应文本插入到 span 元素中
$("message_a").load("userLogin.do".{username:$(this).val()}).css("color","gray");});
$(document.frm_register).submit(function(){  //设置表单 form1 的 submit 事件处理程序
var status=true;
if($("#password").val()==""{          //若未输入密码
$("#message_b").html("密码不能为空!").css("color","red");  //设置 span 的内容和颜色
$("#password").focuse();              //将焦点转移到 password 密码框
stause=false;
}else{
$("#message_b").html("");
}
if($("#username").val()==""){         //若未输入用户名
$("#message_a").html("用户名不能为空!").css("color","red");  //设置 span 的内容和颜色
$("#password").focuse();              //将焦点转移到 username 密码框
stause=false;
}else{
$("#message_a").html("");
}
this.action="JavaScript;alert('用户名'+$('#username).val());"
return status;                        //若返回 true, 则允许提交表单,否则取消提交表单
});
});
</script>
```

用 jQuery 编写的事件处理程序已经与相应的网页元素绑定,无须再对网页元素进行任

何修改。user_reglster.htm 网页由于引入了 jQuery，书写的代码比只使用 JavaScript 脚本少几十行，jQuery 的简洁、高效由此可见。

下面对这个例子中用到的 jQuery 功能做简要的说明。

1）创建 jQuery 对象

jQuery 对象是指使用 jQuery 工厂函数包装 DOM 元素所产生的对象 jQuery。工厂函数是 jQuery 的核心函数，其名称为 jQuery，简写形式为 $ 。

使用函数 $ 可将 DOM 元素转化为 jQuery 对象。在本例中使用 $(document)将 DOM 元素 document 转化为 jQuery 对象，并调用 ready 方法指定 DOM 加载就绪时执行的一个匿名函数：

```
$(document).ready(function(){
//...
};
```

在本例中还使用 $(document.frm_register)将表单 frm_register 转换成一个 jQuery 对象，并通过对其调用 submit 方法来设置提交表单时执行的事件处理程序。

工厂函数 $ 还可以接收一个包含 $("#username")从方法中获取 id 为 username 的 input 元素并返回一个 jQuery 对象，通过对其调用 blur 的方法来设置文本框的 blur 事件处理程序。

$("#username")与 document.getElementById("username")具有类似的功能，它们都可以从文档中选择出 id 为 username 的 input 元素，但前者是一个 jQuery 对象，后者是一个 DOM 对象。

在编写 jQuery 代码时，正确区分 jQuery 对象和 DOM 对象是十分重要的。

对于 jQuery 对象可以使用 jQuery 特有的各种方法进行操作，但不能对其使用 DOM 属性和方法。同样，对于 DOM 对象可以对其使用 DOM 属性和方法，但不能使用 jQuery 方法。

2）链式操作

在使用 jQuery 工厂函数创建 jQuery 对象之后，即可使用 jQuery 提供的各种方法对其进行操作。大多数 jQuery 方法都具有一个特点，即在对 jQuery 对象执行所需操作后仍然返回此对象本身，从而允许调用另一个 jQuery 方法对它进行操作。在这种操作链条中可以包含任意数目的方法调用，因此使用一个语句就能够完成对同一个对象的多种操作。

在本例中使用以下语句对 id 为 message_a 的 span 元素进行操作，先设置其开始标记与结束标记之间的内容，然后对文本颜色进行设置。

```
$("#message_a").html("用户名不能为空!").css("color","red");
```

其中，$("#message_a")用于从文档中查找 id 为 message_a 的元素并返回一个 jQuery 对象，通过对该 jQuery 对象调用 html 方法来设置匹配元素的 HTML 内容，然后再次返回该对象；接着通过对该 jQuery 对象调用 CSS 方法在所匹配元素中设置 CSS 样式属性 color 的值，这是一个典型的链式操作，它的功能相当于下面两个 JavaScript 语句：

```
document.getElementById("message_a").innerHTML = "用户名不能为空!";
document.getElementById("message_a").style.color = "red";
```

在使用链式操作时，对于同一个对象的多个操作，通常可以写成一行；但考虑到代码的可读性，每行代码最好不要超过 3 个操作，也可以按照功能块将代码划分成多行，而且每行都可以添加注释。例如：

```
$("#message_a").html("用户名不能为空")          //设置 span 元素的内容
.css("color","red");                            //设置 span 元素的文本颜色
```

15.2 jQuery 选择器

选择器是 jQuery 最基础的东西，它的功能可谓强大无比。

15.2.1 jQuery 选择器的概念

选择器允许用户对元素组或单个元素进行操作，在 jQuery 中，对事件处理、遍历 DOM 和 Ajax 操作都依赖于选择器。

语法：

```
$("参数")
```

说明：

根据选择器的不同类型，所带参数也不同。例如：

```
var jQueryObject = $("#testDiv");
```

就是一个 ID 选择器，选择 ID 为 testDiv 的 DOM 对象，并将它放入 jQuery 对象，最后返回了一个 jQuery 对象。

15.2.2 jQuery 选择器的分类

jQuery 选择器按照功能和类型分为几类，如表 15-1 所示。

表 15-1 jQuery 选择器的分类

名　称	中文名称
Basic	基础选择器
Hierarchy	层次选择器
Basic Filters	基本过滤器
Content Filters	内容过滤器
Visibility Filters	可见性过滤器
Attribute Filters	属性过滤器
Child Filters	子元素过滤器
Forms	表单选择器
Forms Filters	表单过滤器

jQuery 选择器可以分为两大类，即“选择”和“过滤”。选择的作用是选择元素，过滤的作用是从选中的元素中筛选元素，如果只使用过滤器，相当于先选中所有元素然后过滤。

15.2.3 基础选择器

基础选择器是最简单的选择器,其中 ID 选择器、元素选择器都是 JavaScript 原生支持的功能,所以在 jQuery 内部可以直接调用,效率较高。

表 15-2 所示为所有的基础选择器。

表 15-2 基础选择器

名 称	中文名称	说 明	举 例
#id	ID 选择器	根据元素 ID 选择	$("divld"):选择 ID 为 divld 的元素
element	元素选择器	根据元素的名称选择	$("a"):选择所有<a>元素
. class	样式选择器	根据元素的 CSS 类选择	$(". bgRed"):选择所用 CSS 类为 bgRed 的元素
*	全选择器	选择所有元素	$(" * "):选择页面中的所有元素
selector1, selector2, selectorN	多重选择器	可以将几个选择器用","分隔,然后拼成一个选择器字符串,选择时会同时选中这几个选择器匹配的内容	$("#divld,a,. bgRed")

基础选择器是最基础且最常用的选择器。根据选择器的优化原则,应该优先使用 ID 选择器和 element 选择器缩小查找范围。如果只是用过滤器,则相当于使用了" * "选择器在所有元素中过滤,应该尽量避免。

15.2.4 层次选择器

层次选择器用于带有层次关系的对象选择。在进行页面开发时大家经常遇到获取一个元素中的某些元素或者获取相邻节点元素的使用场景的情况,此时层次选择器就可以帮助用户完成任务。

表 15-3 所示为所有的层次选择器。

表 15-3 层次选择器

名 称	常用中文名	说 明	举 例
ancestor descendan	后代选择器	使用"form input"的形式选中 form 中的所有 input 元素,即 ancestor(祖先)为 form,descendant(子孙)为 input	$("bg. Reddiv"):选择 CSS 类为 bgRed 的元素中的所有<div>元素
parent>child	子代选择器	选择 parent 的直接子节点 child,必须包含在 parent 中并且直接父类是 parent 元素	$(". myList>li"):选择 CSS 类为 myList 的元素中的直接子节点 对象
prey+next	层次选择器	prey 和 next 是两个同级别的元素,选中在 prey 元素后面的 next 元素	$("#hibiscus+img"):选择 ID 为 hibiscus 的元素后面的 img 对象
prev~ siblings	层次选择器	选择 prey 后面的根据 siblings 过滤的元素,注意 siblings 是过滤器	$("#someDiv~[title]"):选择 ID 为 someDiv 的对象后面带有 title 属性的所有元素

“ancestor descendant”后代选择器是最常用的选择器，但是大家要注意后代选择器和子代选择器的区别，很多时候使用后代选择器和子代选择器的结果是相同的，但是也有例外。

15.2.5 基本过滤器

过滤器和选择器是有区别的，过滤器的作用是在已经选择的元素中进行“过滤”操作，因为单独使用过滤器，相当于使用了“*”这个全选择器，性能会降低，所以在使用过滤器时一定要在过滤器前添加选择器。

表 15-4 所示为所有的基本过滤器。

表 15-4 基本过滤器

名　称	说　明	举　例
:first	匹配找到的第 1 个元素	查找表格的第 1 行：$("tr:first")
:last	匹配找到的最后一个元素	查找表格的最后一行：$(tr:last")
:not(selector)	去除所有与给定选择器匹配的元素，在 jQuery 1.3 中已经支持复杂选择器了(例如 not(diva)和 not(div,a))	查找所有未选中的 input 元素：$ ("input:not(:checked)")
:even	匹配所有索引值为偶数的元素，从 0 开始计数	查找表格的 1、3、5 等行：$("tr. even")
:odd	匹配所有索引值为奇数的元素，从 0 开始计数	查找表格的 2、4、6 等行：$("tr. odd")
:eq(index)	匹配一个给定索引值的元素，index 从 0 开始计数	查找第 2 行：$("tr:eq(1)")
:gt(index)	匹配所有大于给定索引值的元素，index 从 0 开始计数	查找第 2 行、第 3 行，即索引值是 1 和 2，也就是比 0 大：$("tr:gt(0)")
:lt(index)	选择索引小于 *N* 的 elements，index 从 0 开始计数	查找第 1 行、第 2 行，即索引值是 0 和 1，也就是比 2 小：$("tr. lt(2)")
:header	选择所有 h1、h2、h3 类的 header 标签	给页面内的所有标题加上背景色：$(":header"). css("background","#EEE");
:animated	匹配所有正在执行动画效果的元素	只对不执行动画效果的元素执行一个动画特效：$("#run"). click(function(){ $(" div: not(:animated)"). animate({left:"+=20"}, 1000);});

15.2.6 内容过滤器

表 15-5 所示为所有的内容过滤器。

表 15-5 内容过滤器

名 称	说 明	举 例
:contains(text)	匹配包含给定文本的元素	查找所有包含“Tom”的 div 元素：$("div:contains(Tom')")
:empty	匹配所有不包含子元素或者文本的空元素	查找所有不包含子元素或者文本的空元素：$("td:empty")
:has(selector)	匹配含有选择器所匹配元素的元素	给所有包含 p 元素的 div 元素添加一个 text 类：$("div:has(p)").addClass("text")
:parent	匹配含有子元素或者文本的元素	查找所有含有子元素或者文本的 td 元素：$("td:parent")

15.2.7 可见性过滤器

表 15-6 所示为所有的可见性过滤器。

表 15-6 可见性过滤器

名 称	说 明	举 例
:hidden	匹配所有的不可见元素	查找所有 display 为 none 的 tr 元素：("tr:hidden")
:visible	匹配所有的可见元素	查找所有 display 不为 none 的 tr 元素：$("tr:visible")

15.2.8 属性过滤器

表 15-7 所示为所有的属性过滤器。

表 15-7 属性过滤器

名 称	说 明	举 例
[attribute]	匹配包含给定的属性的元素	查找所有含有 ID 属性的 div 元素：$("div[id]")
[attribute=value]	匹配给定的属性是某个特定值的元素	查找所有 name 属性是 newsletter 的 input 元素：$("input[name='newsletter']").ttr("checked",true)
[attribute!=value]	匹配给定的属性是不包含某个特定值的元素	查找所有 name 属性不是 newsletter 的 input 元素：$("input[name!='newsletter']").attr("checked",true)
[attribute^=value]	匹配给定的属性是以某些值开始的元素	$("input[name='news']")
[attribute$=value]	匹配给定的属性是以某些值结尾的元素	查找所有 name 以'letter'结尾的 input 元素：$("input[name$='letter']")
[attribute*=value]	匹配给定的属性是以包含某些值的元素	查找所有 name 包含'little'结尾的 input 元素：$("input[name*='letter']")
[attributeFilter1] [attributeFilter2] [attributeFilterN]	复合属性选择器，在同时满足多个条件时使用	找到所有含有 ID 属性，并且它的 name 属性是以 man 结尾的元素：$("input[id][name$='man']"

属性过滤器是最常使用的过滤器，使用属性过滤器甚至可以完成许多其他过滤器的功能。

15.2.9 表单选择器

表单选择器在 jQuery 官方的分类是“Form”。表 15-8 所示为所有的表单选择器。

表 15-8 表单选择器

名　称	说　明	举　例
:input	匹配所有 input、textarea、select 和 button 元素	查找所有 form 元素：$(":input")
:text	匹配所有文本框	查找所有文本框：$(":text")
:password	匹配所有密码框	查找所有密码框：$(":password")
:radio	匹配所有单选按钮	查找所有单选按钮：$(":radio")
:checkbox	匹配所有复选框	查找所有复选框：$(":checkbox")
:submit	匹配所有提交按钮	查找所有提交按钮：$(":submit")
:image	匹配所有图像域	匹配所有图像域：$(":image")
:reset	匹配所有重置按钮	查找所有重置按钮：$(":reset")
:button	匹配所有按钮	查找所有按钮：$(":button")
:file	匹配所有文件域	查找所有文件域：$(":file")

15.2.10 子元素过滤器

子元素过滤器用于从子对象集合中过滤。表 15-9 所示为所有的子元素过滤器。

表 15-9 子元素过滤器

名　称	说　明	举　例
:nth-child(index/even/odd/equation)	匹配其父元素下的第 *N* 个子或奇偶元素。':eq(index)'只匹配一个元素，而这个选择符将为每一个父元素匹配子元素。 :nth-child 从 1 开始，而:eq 是从 0 算起的。可以使用： :nth-child(even) :nth-child(odd) :nth-child(3n) :nth-child(2) :nth-child(3n+1) :nth-child(3n+2)	在每个 ul 查找第 2 个 li：$("ul li: nth-child(2)")
:first-child	匹配第一个子元素，':first'只匹配一个元素，而此选择符将为每个父元素匹配一个子元素	在每个 ul 查找第 1 个 li：$("ul li:first-child")
:last-child	匹配最后一个子元素，':last'只匹配一个元素，而此选择符将为每个父元素匹配一个子元素	在每个 ul 查找最后一个 li：$("ul li:1ast-child(2)")
:only-child	如果某个元素是父元素中唯一的子元素，将会被匹配，如果父元素中含有其他元素，则不会被匹配	在 ul 查找唯一子元素的 li：$("ul li:only-child")

子元素过滤器并不经常使用，因为通常需要过滤的是具有某些业务逻辑的对象，在使用时需要注意 nth-child 过滤器是从 1 开始的，而 eq 过滤器是从 0 开始的。

15.2.11 表单过滤器

表单过滤器根据表单的属性(例如“可用”、“不可用”、“选中”等状态值)对选中的集合进行过滤。表 15-10 所示为所有的表单过滤器。

表 15-10 表单过滤器

名 称	说 明	举 例
:enabled	匹配所有可用元素	查找所有可用的 input 元素： $("inpu:enabled")
:disabled	匹配所有不可用元素	查找所有不可用的 input 元素： $("input:disabled")
:checked	匹配所有被选中元素	查找所有选中的复选框元素： $("input:checked")
:selected	匹配所有选中的 option 元素	查找所有选中选项的元素： $("select option:selected")

15.3 查找与筛选元素

jQuery 的筛选函数提供了串联、查找和过滤函数，为用户的 jQuery 对象操作带来了很大的方便。

15.3.1 过滤函数

过滤函数的作用是在已经选定的集合中将与过滤函数匹配的元素保留，不匹配的则去除。相关的过滤函数如表 15-11 所示。

表 15-11 过滤函数

名 称	说 明	示 例
eq(index)	返回集合中索引为 index 的元素	$("#dtr").eq(2)：返回匹配的第 3 个元素
filter(expr)	按指定表达式筛选元素	$("#d").filter(".checked")：返回持有“.checked”类的元素
filter(fn)	与指定函数返回值匹配的元素集合	$("td").filter(function(){return $(this).css('eolor')=='red'; }).ess("background-color","green")； 将表格 td 中的文字颜色由红色转为绿色
is(expr)	待匹配的集合元素中只要存在一个使得 expr 为 true，则返回真，否则返回假	$("input"[type='text']).parente(.)is('form') 上述表达式返回真值

续表

名　称	说　明	示　例
map(callback)	将一组元素转为数组形式	$("#d").append($("input").map(funetion(){ return $(this).val();})).get().join(":"); 将表单中 input 的值使用":"连接为值的列表
not(expr)	从筛选结果中去除不匹配的颜色	$("tr").not($("tr:eq(0)")).css("background-color","gray"); 将表格中从第 2 行到结束行的背景颜色改为"gray"
slice(start,end)	在集合元素中截取	$("tr").slice(0,2):选择 tr 的前两个元素

15.3.2　查找函数

查找函数的作用是从集合中再次查找匹配元素，相关查找函数如表 15-12 所示。有时过滤函数和查找函数没有明显的分界。

表 15-12　查找函数列表

名　称	说　明	示　例
add(expr)	expr 可以是用于匹配元素的选择器字符串、DOM 对象、jQuery 对象或者 HTML 字符串	$("#d").add("<label>demo</label>") 构建一个 label 并添加到匹配元素中
children([expr])	选择与 expr 匹配的直接子元素，如果省略 expr 则选择所有的直接子元素	$("#form1").children() 选择表单 forml 的所有直接子元素
contents()	返回匹配元素内的所有子节点(包含文本节点)	$("#d#form1").contents().css('color','red') 将匹配元素的所有直接子元素的文字颜色设置为"red"
find(expr)	返回所有与指定表达式匹配的元素	$("#d").find("label")与$("d label")的作用相同
next(expr)	返回与指定表达式匹配的每个元素的相邻元素集合	$("input[type='submit']").next() 选择与类型为"submit"的 input 相邻的兄弟元素
nextAll(expr)	返回与指定表达式匹配的元素的所有同辈元素集合	$("#tdl").nextAll() 选择匹配元素的所有同辈元素
parent([expr])	返回一个匹配表达式的所有元素的共同且唯一的父元素	$("#forml").parent() 选择匹配元素的父元素
parents([expr])	返回一个包含匹配元素的祖先元素的元素集合	$("input").parents() 选择匹配元素的所有祖先元素
prev([expr])	返回与指定表达式匹配的元素集合中的每个元素之前的所有元素集合	$("p").prev() 选择匹配元素的相邻的前一个同辈元素集合
prevAll([expr])	查找匹配表达式元素之前的所有同辈元素	$("#commandQuery").prevAll() 选择匹配元素之前的所有同辈元素
siblings([expr])	返回一个包含匹配元素集合中每一个元素的所有唯一同辈元素的元素集合	$("#d").siblings() 选择匹配元素的所有同辈元素

15.3.3 用 jQuery 操作 DOM

在 jQuery 中，attributes 函数用来操作 DOM 元素的属性和样式。一些函数会出现在多种分类中，attributes 函数如表 15-13 所示。

表 15-13 attributes 函数列表

函数名称	函数说明
.addClass	为选中的元素添加样式
.attr	获取或者设置元素的属性
.hasClass	判断选中的元素是否包含了指定的样式
.html	获取或者设置元素的 HTML 内容
.removeAttr	为选定的对象移除指定的属性
.removeClass	为选定的对象移除指定的样式
.text	获取或者设置元素的文字内容
.var	获取或者设置元素的值

与操作元素属性相关的函数如表 15-14 所示。

表 15-14 attr 函数的用法

函数名称	函数说明	返回值
.attr(attrName)	取得第一个匹配元素的属性值，如属性不存在返回 undefined	字符串
.attr(attrName. value)	为所有匹配的元素设置属性值	无
.attr(map)	将一个"键/值"对形式的对象为所有匹配元素设置属性和值	无
.removeAttr(attrName)	移除匹配元素的属性	无

与操作元素 CSS 相关的函数如表 15-15 所示。

表 15-15 CSS 函数

函数名称	函数说明	返回值
.addClass(className)	为匹配的元素增加指定的样式	无
.removeClass(className)	为匹配的元素移除指定的样式	无
.hasClass(className)	判定选中的元素是否包含指定的样式	布尔
.css(propName)	返回第一个匹配元素指定的 CSS 属性值	字符串
.css(propName. value)	为匹配的元素设置指定的 CSS 属性值	无
.css(map)	将一个"键/值"对形式的对象为所有匹配元素设置 CSS 属性	无
.height()	获取选中元素的高度	数值
.height(val)	设置选中元素的高度	无
.weight()	获取选中元素的宽度	数值
.weight(val)	设置选中元素的宽度	无
.innerHeight([val])	获取/设置选中元素内部区域的高度	数值/无
.innerWeight([val])	获取/设置选中元素内部区域的宽度	数值/无
.outerHeight([val])	获取/设置选中元素外部区域的高度	数值/无
.outerWeight([val])	获取/设置选中元素外部区域的宽度	数值/无
.position([val])	获取/设置选中元素相对于父元素的偏移	数值/无
.offset([val])	获取/设置选中元素相对当前窗口的偏移	数值/无
.toggleClass(className)	如果存在(不存在)就删除(添加样式)	无

jQuery 还提供了 html()和 text()函数改变元素的内容,如表 15-16 所示。

表 15-16　html()和 text()函数

函数名称	函数说明	返回值
.html()	取得第一个匹配元素的 HTML 内容	字符串
.html(htmlStr)	设置所有匹配的元素的 HTML 内容	无
.text	取得第一个匹配元素的文字内容	字符串
.test(str)	设置所有匹配元素的文字内容	无

15.4　jQuery 工具函数

工具函数是指在 jQuery 对象(即变量"$")上定义的函数。

15.4.1　浏览器特性检测

由于历史原因,各种不同核心浏览器的不同版本支持的功能也不尽相同。出于对不同浏览器的兼容性的考虑,需要检查当前浏览器的版本,从而做出相应的处理。

jQuery 检查浏览器类型、浏览器版本,浏览器支持的属性有以下几种:

1. jQuery.support

jQuery.support 是一组用于展示不同浏览器各自特性和 bug 的属性集合,其检测的属性如表 15-17 所示。

表 15-17　jQuery.support 检测的属性

属性名称	说明
boxModel	如果这个页面和浏览器是以 W3C CSS 盒式模型来渲染的,则等于 true。通常在 IE 6 和 IE 7 的怪癖(Quriks Mode)模式中,这个值是 false。在 document 准备就绪前,这个值是 null
cssFloat	如果用 cssFloat 来访问 css 的 float 的值,则返回 true。目前,在 IE 中会返回 false,用 styleFloat 代替
hrefNormalized	如果浏览器从 getAttribute("href")返回的是无变化的结果,则返回 true。在 IE 中会返回 false
htmlSerialize	如果浏览器通过 innerHTML 插入链接元素的时候会序列化这些链接,则返回 true。目前,在 IE 中会返回 false
noCloneEvent	如果浏览器在复制元素的时候不会连同事件处理函数一起复制,则返回 true。目前,在 IE 中会返回 false
objectAll	如果在某个元素对象上执行 getElementsByTagName("*")会返回所有子孙元素,则为 true。目前,在 IE 7 中为 false
scriptEval	使用 append Child/createTextNode 方法插入脚本代码时浏览器是否执行脚本,在 IE 中返回 false
style	如果 getAttribute("style")返回元素的行内样式,则为 true。在 IE 中为 false
tbody	如果浏览器允许 table 元素不包含 tbody 元素,则返回 true。目前,在 IE 中会返回 false,IE 会自动插入缺失的 tbody

代码示例：

```
if(jQuery.support.tbody){alert("包含 tbody 元素");}
```

2. jQuery.browsername

jQuery.browsername 用于检查是哪种浏览器，目前支持 4 种主流浏览器：

（1）IE：msie。

（2）Firefox：mozilla。

（3）Opera：opera。

（4）Safari：safari。

代码示例：

```
if($.browser.safari){alert("this is safari!");}
```

3. jQuery.browser.version

jQuery.browser.version 用于显示浏览器的版本号。

4. jQuery.boxModel

jQuery.boxModel 用于检查当前页面中浏览器是否使用标准盒模型渲染页面，建议使用 jQuery.support.boxModel 代替 W3C CSS 盒模型。

在 Internet Explorer 怪癖(Quirks Mode)模式中返回 false。

代码示例：

```
$.boxModel
```

15.4.2 数组和对象操作

数组是非常重要的概念，在 jQuery 中对此提供了一系列的函数，数组的相关函数如表 15-18 所示。

表 15-18 jQuery 数组的相关函数

函数名称	函数说明	返回值
.each(obj, callback(index, val))	可用于遍历任何对象。回调函数拥有两个参数，第一个为对象的成员或数组，第二个为对应变量或内容。如果需要退出 each 循环，可以使回调函数返回 false	对象
.extend(target, [obj1], [objn])	用一个或多个其他对象来扩展一个对象，返回被扩展的对象。如果不指定 target，则对 jQuery 命名空间本身进行扩展。obj1～objn 为将合并到 target 的对象	对象
.grep(array, fn(element, index, [invert]))	使用过滤函数过滤数组元素。此函数至少传递两个参数，即待过滤数组和过滤函数，过滤函数必须返回 true 以保留元素或返回 false 以删除元素。如果“invert”为 false 或未设置，则函数返回数组中由过滤函数返回 true 的元素，当“invert”为 true 时，则返回过滤函数中返回 false 的元素集	数组

续表

函数名称	函数说明	返回值
.makeArray(obj)	将类数组对象转换为数组对象。类数组对象有 length 属性,其成员索引为 0 至 length-1。实际上此函数在 jQuery 中自动使用而无须特意转换。obj 为类数组对象	数组
.map(array. callback(elemer, index))	将一个数组中的元素转换到另一个数组中。作为参数的转换函数会为每个数组元素调用,而且会给这个转换函数传递一个表示被转换元素作为参数。转换函数可以返回转换后的值、null(删除数组中的项目)或一个包含值的数组,并扩展至原始数组中	数组
.inArray(value,array)	确定第 1 个参数在数组中的位置,从 0 开始计数(如果没有找到则返回－1)	数值
.isArray(obj)	测试对象是否为数组	布尔
.merge(first,second)	合并两个数组。返回的结果会修改第 1 个数组的内容——第 1 个数组的元素后面跟着第 2 个数组的元素	数组
.unique(array)	除数组中的重复元素,只处理删除 DOM 元素数组,不能处理字符串或者数字数组	数组
.parseJSON(json)	接受一个 JSON 字符串,返回解析后的对象。传入一个非法的 JSON 字符串会抛出异常	字符串

15.4.3 其他工具函数

jQuery 还包含很多实用的工具函数,在实际使用中最常用的莫过于字符串去空格操作及判断函数,如表 15-19 所示。

表 15-19 常用的其他工具函数

函数名称	函数说明	返回值
.trim(str)	去掉字符串起始和结尾的空格	字符串
.isEmptyObject(obj)	测试对象是否为空对象(不包含任何属性)	布尔
.isFunction(obj)	测试对象是否为函数	布尔
.isPlainObject(obj)	测试对象是否为纯粹的对象(通过"{}"或者"new Object"创建的)	布尔
.isNumeric(val)	测试对象是否为一个数字	布尔
.isWindow(obj)	测试对象是否为窗口(有可能是 Frame)	布尔

15.5 jQuery UI

jQuery UI 是以 jQuery 为基础的开源 JavaScript 网页用户界面代码库,包含底层用户交互、动画、特效和可更换主题的可视控件。

15.5.1 jQuery UI 基础

jQuery UI 是 jQuery 的官方插件,是在 jQuery 之上开发的专门用于 UI 交互的类库的,使用 jQuery UI 最直接的方法就是使用它提供的 UI 控件,例如日历框、弹出框等。另外,还

可以基于 jQuery UI 提供的底层控件接口(例如可拖曳、可拉伸等)来开发控件。

jQuery UI 提供了丰富的“皮肤”(“主题”),所以可以快速地更换控件的样式。jQuery UI 的在线网站十分强大,用户在下载时可组装想要的功能定制下载。

jQuery UI 包括下面 4 个部分。

(1) Interacttions(交互):Interactions 表示提供底层的用户交互特效。例如可以为一个 div 添加控件拖动功能、重置大小功能。可以说这是一组底层特效接口,用户可以在这些特效的基础上开发控件。

(2) Widgets(页面控件):Widgets 是在 Interactions 的基础上已经由 jQuery UI 开发好的可以快速使用的用户页面控件,例如弹出框控件、Tab 控件、日历控件等。

(3) Efffects(页面效果):Effects 是指 easmg 特效,在 jQuery UI 中提供了各种不同的 easing()函数及各种定义好的动画特效。

(4) Utilities(工具集):Utilities 提供了底层的工具函数,用来实现用户交互、各种控件及效果。

15.5.2 Datepicker 控件

日历控件是每一个 Web 应用都不可或缺的控件。jQuery UI 提供了日历控件 Datepicker。日历控件 Datepicker 依赖 jQuery UI 的 UI Core 组件部分。Datepicker 控件的可配置性和扩展性非常高,例如可以配置日期格式、语言、日期范围等,并且提供了各种事件和方法,便于用户加入业务逻辑,Datepicker 还支持以下键盘操作。

- page up/down:更改月份。
- ctrl+page up/down:更改年份。
- ctrl+home:回到本月。
- ctrl+left/right:选择上一天或下一天。
- ctrl+up/down:选择上一周或下一周。
- enter:选中日期。
- ctrl+end:关闭日历并清空选中日期。
- esc:关闭日历不选择日期。

1. 属性

日历控件的属性说明如表 15-20 所示。

表 15-20 日历控件 Datepicket 的属性说明

参数名称	数据类型	默 认 值	说 明
disabled	Boolean	false	通过此属性可以控制日历框是否有效
altField	String	""	另外一个元素的选择器表达式,在选中日期时使用 altFormat 属性定义的日期格式,将选中日期放入此元素

续表

参数名称	数据类型	默　认　值	说　　明
altFormat	String	""	填充 altField 表达式选中元素的日期格式
appendText	String	""	在日期控件的页面元素后面加入一段文字。经测试是加入了一段 span，例如： $(". selector"). datepicker({appendText:'(yyyy-mm-dd)'})； 加入的 HTML 为： (yyyy-mm-dd)
autoSize	Boolean	false	是否自动设置 input 元素的大小，如果设置为 true，将根据 dateFormat 属性格式化后的日期字符串重置 input 元素的大小
buttonImage	String	""	弹出按钮图片的 URL 地址。如果设置了此属性，则 buttonText 属性会包含此图片的 alt 和 title 属性
buttonImageOnly	Boolean	false	在默认情况下，如果将"showOn"属性设置为"button"或"both"，将在 input 后添加一个元素，也可能用来显示日历。默认添加 buttonImage 会在 button 内部添加一张图片，如果将此属性设置为 true，则会将 button 元素变成 img 元素
button Text	String	"..."	指定触发按钮上显示的文本
calculate Week	Function	S. datepicker. iso8601 Week	根据日期计算本周是第几周的函数，它是根据 ISO8601 标准实现的，每周一是一周的开始，每年的第一周是包含本年第一个星期四的一周
changeMonth	Boolean	false	true 表示允许从下拉框更改月份
changeYear	Boolean	false	true 表示允许从下拉框更改年份
closeText	String	"Done"	关闭按钮的文件，这是一个区域化属性，例如如果选择了中文语言包，则会使用此语言包的此属性，需要在使用"showButtonPanel"属性显示关闭按钮时此属性才起作用
constrainInput	Boolean	true	如果设置为 true，则为 input 中 dateFormat 定义的格式中允许的字符(只限制字符不限制格式)
currentText	String	"Today"	"今天"按钮显示的文字，与"closeText"属性一样是区域化的文字之一，需要配合"showButtonPanel"属性使用

续表

参数名称	数据类型	默 认 值	说 明
dateFormat	String	"mm/dd/yy"	日期格式文字串,控制选中日期后填充的格式,它是区域化属性之一,例如中文语言包会自动改成“yy-mm-dd”
dayNames	Array	['Sunday','Monday','Tuesday','Wednesday','Thursday','Friday','Saturday']	每天是星期几的文字,它是区域属性之一,从星期日开始。此文字是星期的全名,会出现在选中后的 input 文字中,以及当鼠标停留在日历框的 dayNamesMin(星期简写)上时显示的文字。使用中文语言包会设置为['星期日','星期一','星期二','星期三','星期四','星期五','星期六']
dayNamesMin	Array	['Su','Mo','Tu','We','Th','Fr','Sa']	星期最简单的缩写,在弹出框上会显示此文字,它是区域属性之一,中文语言包会设置为['日','一','二','三','四','五','六']
dayNamesShort	Array	['Sun','Mon','Tue','Wed','Thu','Fri','Sat']	星期的缩写。和 dayNameMin 的不同之处是它用在选中日期后填充 Input 时使用。中文语言包会设置为['周日','周一','周二','周三','周四','周五','周六']
defaultDate	Date,Number,String	null	如果 input 没有日期,则日历框打开后会高亮显示 defaultDate 中设置的日期,此高亮是鼠标悬浮在某日期上时的高亮效果,而不是选中日期的高亮效果。如果直接按下回车,则会在 input 中填充此日期
duration	String,Number	"normal"	日历框动画效果的渐变时间
firstDay	Number	0	设置每周的第一天是星期几,0 表示星期日,它是区域属性,中文语言包中将此值设置为 1
gotoCurrcnt	Boolean	false	如果设置为 true,“回到当前日期”连接将改为回到选中的日期,默认为回到今天
hideIfNoPrevNext	Boolean	false	如果设置为 true,则当超过了 minDate 和 maxDate 设置的日期时上一个月和下一个月的链接将不显示
isRTL	Boolean	false	是否为“Right To Left”(右到左),有的语言是从右到左阅读的
maxDate	Date,Number,String	null	可以选择的最大日期
minDate	Date,Number,String	null	可以选择的最小日期

续表

参数名称	数据类型	默 认 值	说 明
monthNames	Array	[' January ' , ' February ' , 'March', 'April', 'May', 'June', ' July', 'August', 'September', ' October ' , ' November ' , 'December']	每个月的全名，它是区域属性，中文语言包为['一月','二月','三月','四月','五月','六月','七月','八月','九月','十月','十一月','十二月']
monthNamesShort	Array	['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']	每个月的缩写，它是区域属性，中文语言包为['一','二','三','四','五','六','七','八','九','十','十一','十二']
navigationAsDate Format	Boolean	false	如果设置为 true，则 formatDate()函数会作用在 nextText、prevText 和 currentText 属性上，允许它们使用日期作为文字内容
nextText	String	"Next"	下一个月按钮的提示文字
numberOfMonths	Number，Array	1	一次显示几个月份。如果传递数字，则表示在一行中显示几个月份，也可以传递一个含有两个元素的数组。例如，[2,3]表示显示两行，每行显示 3 个月份
prevText	String	"prev"	上一个月按钮的提示文字
selectOtherMonths	Boolean	false	只有当 selectOtherMonths 被设置为 true 时才生效，在当月显示的其他月份的日期可以被选中
shourt YearCutoff	String，Number	"+10"	设置截止年份的值。如果是 0～99 的数字，则从当前年份开始算起；如果为字符串，则相应地转换为数字后再与当前年份相加；当超过截止年份时，则被认为是上一个世纪
showAnim	String	"show"	设置显示、隐藏日期插件的动画的名称
showCurrentAtPos	Number	0	设置在多月份显示的情况下当前月份显示的位置，自顶部左边开始第 *X* 位
show Month After Year	Boolean	false	是否在面板的头部年份后面显示月份
showOn	String	"focus"	设置什么事件触发显示日期插件的面板，可选值有 focus、button 和 both
showOptions	Options	{}	如果使用 showAnim 显示动画效果，可以通过此参数增加一些附加的参数设置

续表

参数名称	数据类型	默 认 值	说 明
showOtherMonths	Boolean	false	是否在当前面板显示上、下两个月的一些日期数
stepMonths	Number	1	当单击上/下一月时一次翻几个月
yearRange	String	"－10:＋10"	控制年份的下拉列表中显示的年份数量，可以是相对当前年，也可以是绝对值

2. 方法

1）实例方法

日期选择器也有 destroy、disable、enable 和 option 方法。

```
$(selector).datepicker("isDisabled");     //是否被禁用
$(selector).datepicker("hide",[speed]);   //关闭选择器,speed 为速度
$(selector).datepicker("show");           //调用先前的选择器
$(selector).datepicker("getDate");        //获取选择器中的当前日期
$(selector).datepicker("setDate",date);   //设置当前日期可以为 02/26/2011 或 + 1m + 7d 等
```

2）全局方法

```
$.datepicker.setDefaults(Options);         //设置所有选择器的默认属性
$.datepicker.formatDate(format,date,setting);  //format 为字符串,日期格式,Date 为要显示
的日期值,Setting 为可选项,其值是对象
$.datepicker.parseDate(format,value,setting);  //value 为字符串,包含待提取的日期值
```

3. 事件

1）beforeShowDay 事件

当选择器上的每一天显示之前触发此事件。

事件绑定：

```
$(selector).datepicker({(beforeShowDy:function(date){}});
```

date 表示一个日期，该函数必须返回一个数组。[0]：true 或 false 表示是否可选；[1]：表示此日期的 CSS 类名，默认为""；[2]：表示元素是此日期的一个弹出提示（可选）。

2）onChangeMonthYear 事件

当选择器移动到新的年份或月份时触发。

事件绑定：

```
$(selector).datepicker({onChangeMonthYear,function(year,month,inst){ }});
```

year 表示所选的年份；month 表示所选的月份（1～12）；inst 表示日期选择器，它指向所关联的输入控件。

3）onClose 事件

当选择器关闭时触发。

事件绑定：

```
$(selector).datepicker({onClose:funct:ion(dateText,inst){ }});
```

dateText 表示所选择的日期为文本字符串，如未选择，则为""；inst 表示日期选择器实例。

4）onSelect 事件

当选择一个日期时触发。

事件绑定：

```
$(selector).datepicker({onSelect:function(dateText,inst){ }});
```

参数同上。

15.5.3 Button 控件

Button 控件能够将表单元素、各种类型的 input 元素（例如 submit）、a 元素等变成按钮样式显示，并且具有 mouseover 等样式效果。

1. 属性

Button 控件的属性如表 15-21 所示。

表 15-21 Button 控件的属性说明

属性	数据类型	默 认 值	说 明
text	Boolean	true	是否显示文本，若为 false，必须启用图标
Icons	Option	{primary:null secondary:null}	指定显示的图标，属性值为字符串类名，分别为左边的图标和右边的图标
label	String	按钮的 value 属性	按钮上显示的文本

2. 方法

按钮控件也有 disable、enable、option、destroy 等方法。

```
$(selector).button([options]);          //普通按钮的构造方法
$(selector).buttonset();                //单选按钮或复选框的构造方法
```

3. 事件

create 事件的类型为 buttoncreate，当按钮创建时触发。

代码示例：

```
//提供一个回调函数对 enable 事件进行操作
$('.selector').button({create:function(event,ui){...}});
//使用 buttoncreate 类型绑定 create 事件
$('.selector').on("buttoncreate",function(event,ui){});
```

15.6 综合应用

利用 jQuery 制作一个图片放大显示的效果。

(1) 查找素材,并利用 Photoshop 处理图片。

建立 images 文件夹,将素材图片保存在该文件夹下。

(2) 下载并使用 jQuery 编写 JavaScript 客户端脚本文件。

① 下载 jquery.js 文件。

② 编写 photo.js 文件,代码如下:

```
$(document).ready(function(){
$('.photo_slider').each(function(){

var $this = $(this).addClass('photo-area');
var $img = $this.find('img');
var $info = $this.find('.info_area');

var opts = {};

$img.ready(function(){
    opts.imgw = $img.width();
    opts.imgh = $img.height();
});

opts.orgw = $this.width();
opts.orgh = $this.height();

$img.css ({
    marginLeft: "-150px",
    marginTop: "-150px"
});

var $wrap = $('<div class="photo_slider_img">').append($img).prependTo($this);

var $open = $('<a href="#" class="more_info">More Info &gt;</a>').appendTo($this);

var $close = $('<a class="close">Close</a>').appendTo($info);

opts.wrapw = $wrap.width();
opts.wraph = $wrap.height();

$open.click(function(){
    $this.animate({
        width: opts.imgw,
        height: (opts.imgh+95),
        borderWidth: "10"

    }, 600 );
```

```
        $open.fadeOut();

        $wrap.animate({
            width: opts.imgw,
            height: opts.imgh
        }, 600 );

        $(".info_area", $this).fadeIn();

        $img.animate({
            marginTop: "0px",
            marginLeft: "0px"
        }, 600 );

        return false;
    });

    $close.click(function(){
        $this.animate({
            width: opts.orgw,
            height: opts.orgh,
          borderWidth: "1"
        }, 600 );

        $open.fadeIn();

        $wrap.animate({
            width: opts.wrapw,
            height: opts.wraph
              }, 600 );

            $img.animate({
                marginTop: "-150px",
                marginLeft: "-150px"
            }, 600 );

        $(".info_area", $this).fadeOut();
        return false;
    });

    });
    });
```

(3) 制作网页文件 index.html，代码如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/
xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>jQuery应用</title>
<link rel="stylesheet" type="text/css" href="style.css" />
<script type="text/javascript" src="js/jquery.js"></script>
</head>
<body>
<div id="page-wrap">
    <h1>江山四季</h1>
    <table><tr>
    <td><div class="photo_slider">
        <img src="images/spring.jpg"/>
        <div class="info_area"><p> </p>
            <h3>春花</h3>
        </div>
    </div></td>
<td><div class="photo_slider">
<img src="images/summer.jpg"/>
<div class="info_area"><p> </p>
    <h3>夏荷</h3>
</div>
</div></td>
    <td><div class="photo_slider">
      <img src="images/autumn.jpg" />
        <div class="info_area"><p> </p>
            <h3>秋叶</h3>
        </div>
    </div></td>
    <td><div class="photo_slider">
        <img src="images/winter.jpg"/>
        <div class="info_area"><p> </p>
            <h3>冬雪</h3>
        </div>
    </div></td>
    </tr></table>
</div>
<script type="text/javascript" src="js/photo.js"></script>
</body>
</html>
```

(4) 编写CSS文件style.css,代码如下:

```
* { margin: 0; padding: 0; }
body { font: 62.5% "Gill Sans",Georgia, sans-serif;
background: url(images/body-bg.jpg) top center repeat-x white;
}
p { font-size: 1.2em; line-height: 1.2em; }
a { outline: none; color: red;}
a:hover { color: black; }
```

```
a img { border: none; }
h1 { color: black; font-size: 4.0em; }
h3 { font-size: 2.0em; margin-bottom: 5px;}
.clear { clear: both; }
#page-wrap {
margin: 20px;
padding: 10px;
}
a.close {
position: absolute;
right: 10px;
bottom: 10px;
display: block;
width: 20px;
height: 21px;
background: url(images/close_button.jpg);
text-indent: -9999px;
}
.photo_slider_img {
width: 100px;
height: 100px;
margin-bottom: 5px;
overflow: hidden;
}
td {
vertical-align: top;
}
.photo_slider {
position: relative;
width: 100px;
height: 130px;
padding: 10px;
border: 1px solid black;
overflow: hidden;
margin: 25px 10px 10px 10px;
background: white;
float: left;
}
.info_area {
display:none;
}
.more_info {
display: block;
width: 89px;
height: 26px;
background: url(images/moreinfo.jpg);
text-indent: -9999px;
cursor: pointer;
}
```

（5）代码的运行效果如图 15-1 所示。



图 15-1　图片放大显示效果

参 考 文 献

[1] 温俊芹.Flash CS3 动画制作基础与案例教程.北京：北京理工大学出版社，2008.
[2] 闫再华，等.Flash CS5 中文版基础教程.北京：人民邮电出版社，2014.
[3] 赵博，等.从零开始 Photoshop CS5 基础培训教程.北京：人民邮电出版社，2014.
[4] 江可，等.Adobe Photoshop CS3 标准培训教材.北京：人民邮电出版社，2008.
[5] 林跃进，等.Web 前端开发技术与 JavaScript 框架编程.大连：东软电子出版社，2013.
[6] 储良久.Web 前端开发技术.北京：清华大学出版社，2013.
[7] 传智播客高教产品研发部.网页设计与制作(HTML+CSS).北京：中国铁道出版社，2014.
[8] 许莉，等.HTML 与 CSS 前台页面设计.北京：中国水利水电出版社，2013.
[9] 英特尔软件教材学院编写组.HTML5 时代的 Web 应用开发.大连：东软电子出版社，2013.